

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΡΓΑΣΤΗΡΙΟ ΑΝΑΓΝΩΡΙΣΗΣ ΠΡΟΤΥΠΩΝ
ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΗΣ ΤΗΣ
MATLAB

Βαγγέλης Δερματάς

Πάτρα 1999

Οι σημειώσεις αυτές φτιάχτηκαν με την βοήθεια των μεταπτυχιακών φοιτητών του τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών, Διονύση Ανυφαντή, Θανάση Κούτρα, και Ιωάννας Χριστογιάννη.

Βαγγέλης Δερματάς
Λέκτορας

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1	ΤΙ ΕΙΝΑΙ Η MATLAB.....	4
2	ΤΟ ΣΥΣΤΗΜΑ MATLAB.....	5
2.1	Η ΕΚΚΙΝΗΣΗ ΤΗΣ MATLAB.....	6
2.2	ΤΟ ΠΑΡΑΘΥΡΟ ΕΝΤΟΛΩΝ.....	6
2.3	ΔΙΟΡΘΩΣΗ ΕΝΤΟΛΩΝ ΣΤΟ ΠΑΡΑΘΥΡΟ ΕΝΤΟΛΩΝ.....	7
2.4	Η ΕΝΤΟΛΗ FORMAT.....	8
2.5	ΚΑΤΑΣΤΟΛΗ ΕΞΟΔΟΥ.....	9
2.6	ΜΕΓΑΛΕΣ ΓΡΑΜΜΕΣ ΕΝΤΟΛΩΝ.....	9
3	Ο ΧΩΡΟΣ ΕΡΓΑΣΙΑΣ ΤΗΣ MATLAB.....	10
3.1	Άνοιγμα και αποθήκευση των μεταβλητών του χώρου εργασίας.....	10
3.2	Καθορισμός μορφοποίησης αρχείου (FORMAT).....	11
3.3	Διαβάζοντας ASCII αρχεία.....	11
3.4	Αποθήκευση δεδομένων όταν το όνομα αρχείου υπάρχει σε μεταβλητή χαρακτηρισμών.....	11
3.5	Δημιουργία αρχείων M-FILES.....	12
3.6	Διαδρομή αναζήτησης συναρτήσεων της MATLAB.....	13
3.7	Διαχείριση αρχείων και εντολές λειτουργικού συστήματος.....	13
3.8	Εκτέλεση προγραμμάτων από το περιβάλλον εργασίας.....	13
3.9	Το αρχείο εκκίνησης M-FILE.....	14
4	ΜΕΤΑΒΛΗΤΕΣ ΚΑΙ ΣΤΑΘΕΡΕΣ.....	15
4.1	Ειδικές μεταβλητές και σταθερές.....	15
4.2	Δεικτοδότηση και τμήματα πινάκων.....	16
4.3	Ενοποίηση πινάκων.....	16
5	ΤΕΛΕΣΤΕΣ ΠΡΑΞΕΩΝ.....	18
5.1	Αριθμητικοί τελεστές.....	18
	<i>Οι κανόνες προτεραιότητας για τους αριθμητικούς τελεστές είναι.....</i>	<i>18</i>
5.2	Τελεστές σχέσεων.....	18
5.3	Λογικοί τελεστές.....	19
6	ΠΙΝΑΚΕΣ ΚΑΙ ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ.....	20
6.1	Πίνακες στην MATLAB.....	21
6.2	Προσθήκη και αφαίρεση πινάκων.....	22
6.3	Γινόμενο πινάκων και αναστροφος πίνακας.....	23
6.4	Μοναδιαίος πίνακας.....	25
6.5	Το γινόμενο Kronecker tensor.....	25
6.6	Μέτρο διανυσμάτων και πινάκων.....	26
6.7	Επίλυση γραμμικών εξισώσεων.....	27
6.7.1	<i>Εύρεση μοναδικής λύσης.....</i>	<i>27</i>
6.7.2	<i>Αντίστροφοι πίνακες και ριζούσες.....</i>	<i>28</i>
6.7.3	<i>Ψευδοαντίστροφοι πίνακες.....</i>	<i>28</i>
7	ΕΛΕΓΧΟΣ ΡΟΗΣ.....	30
7.1	Εντολές ελέγχου ροής προγράμματος.....	30
8	ΒΟΗΘΕΙΑ ΚΑΙ ΤΕΚΜΗΡΙΩΣΗ.....	34
8.1	Η διαταγή help.....	34
8.2	Παραθύρο help.....	35
8.3	Η διαταγή lookfor.....	35

9	Ο ΔΙΟΡΘΩΤΗΣ ΣΦΑΛΜΑΤΩΝ ΤΗΣ MATLAB	37
9.1	ΔΙΑΔΙΚΑΣΙΑ ΔΙΟΡΘΩΣΗΣ ΣΦΑΛΜΑΤΩΝ	37
9.2	ΈΝΑ ΠΑΡΑΔΕΙΓΜΑ ΔΙΟΡΘΩΣΗΣ ΣΦΑΛΜΑΤΟΣ.....	37
9.2.1	Εκτέλεση του προγράμματος	38
9.2.2	Διόρθωση σφαλμάτων σε Windows95/98/NT	38
9.2.3	Τοποθέτηση σημείων διακοπής εκτέλεσης προγράμματος	39
9.2.4	Έλεγχος μεταβλητών.....	40
9.2.5	Αλλαγή του χώρου εργασίας-Τερματισμός εκτέλεσης συνάρτησης	41
9.2.6	Συνέχιση εκτέλεσης του προγράμματος.....	41
9.2.7	Τερματισμός διαδικασίας διόρθωσης σφαλμάτων	43
9.3	ΔΙΟΡΘΩΣΗ ΠΡΟΓΡΑΜΜΑΤΩΝ ΑΠΟ ΤΗΝ ΓΡΑΜΜΗ ΕΚΤΕΛΕΣΗΣ ΕΝΤΟΛΩΝ	43
9.4	ΠΑΡΑΔΕΙΓΜΑ ΔΙΟΡΘΩΣΗΣ ΠΡΟΓΡΑΜΜΑΤΩΝ ΑΠΟ ΤΟ ΠΑΡΑΘΥΡΟ ΕΚΤΕΛΕΣΗΣ ΕΝΤΟΛΩΝ	43
9.4.1	Καθορισμός σημείων διακοπής προγράμματος.....	43
9.4.2	Μετακίνηση μέσα στο κώδικα	44
9.4.3	Αλλαγή του χώρου εργασίας-Τερματισμός εκτέλεσης συνάρτησης	45
9.4.4	Εμφάνιση m-file με αριθμούς γραμμών	45
9.4.5	Απεικόνιση του καταλόγου εκτελούμενων συναρτήσεων.....	45
9.4.6	Τερματισμός διαδικασίας διόρθωσης προγράμματος.....	47
10	ΑΝΑΓΝΩΣΗ - ΕΓΓΡΑΦΗ ΔΕΔΟΜΕΝΩΝ	48
10.1	ΕΙΣΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ ΣΤΗΝ MATLAB	48
10.2	ΑΠΟΘΗΚΕΥΣΗ ΔΕΔΟΜΕΝΩΝ ΑΠΟ ΤΟ ΠΕΡΙΒΑΛΛΟΝ ΕΡΓΑΣΙΑΣ ΤΗΣ MATLAB ΣΕ ΑΡΧΕΙΑ	49
10.3	ΧΑΡΑΚΤΗΡΕΣ ΔΙΑΧΩΡΙΣΜΟΥ ΔΕΔΟΜΕΝΩΝ	50
11	ΣΥΝΑΡΤΗΣΕΙΣ ΠΟΥ ΔΕΧΟΝΤΑΙ ΟΝΟΜΑΤΑ ΣΥΝΑΡΤΗΣΕΩΝ	51
11.1	ΓΡΑΦΙΚΗ ΠΑΡΑΣΤΑΣΗ ΜΑΘΗΜΑΤΙΚΩΝ ΣΥΝΑΡΤΗΣΕΩΝ	52
12	ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΕ M-FILES	54
12.1	ΔΗΜΙΟΥΡΓΙΑ M-FILES	54
12.2	SCRIPTS	54
12.3	ΣΥΝΑΡΤΗΣΕΙΣ	54
12.4	ΥΠΟΣΥΝΑΡΤΗΣΕΙΣ	57
12.5	ΛΟΓΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ	58
12.6	ΕΚΤΕΛΕΣΗ ΕΝΤΟΛΩΝ ΠΟΥ ΠΕΡΙΕΧΟΝΤΑΙ ΣΕ ΜΕΤΑΒΛΗΤΗ ΧΑΡΑΚΤΗΡΩΝ	59
12.7	ΆΔΕΙΟΙ ΠΙΝΑΚΕΣ	60
12.8	ΕΙΣΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ ΑΠΟ ΤΟΝ ΧΡΗΣΤΗ	60
12.9	ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΤΗΣ ΤΑΧΥΤΗΤΑΣ ΥΠΟΛΟΓΙΣΜΩΝ	60
12.10	ΔΙΑΧΕΙΡΙΣΗ ΜΝΗΜΗΣ	61
13	ΠΙΝΑΚΕΣ ΧΑΡΑΚΤΗΡΩΝ	62
13.1	ΣΥΝΑΡΤΗΣΕΙΣ ΚΕΙΜΕΝΟΥ	63
13.2	ΜΕΤΑΤΡΟΠΗ ΑΠΟ ASCII ΧΑΡΑΚΤΗΡΕΣ ΣΕ ΑΡΙΘΜΗΤΙΚΕΣ ΤΙΜΕΣ ΤΟΥ ASCII ΠΙΝΑΚΑ	64
13.3	ΔΗΜΙΟΥΡΓΙΑ ΔΙΔΙΑΣΤΑΤΩΝ ΠΙΝΑΚΩΝ ΧΑΡΑΚΤΗΡΩΝ	64
13.4	ΣΥΓΚΡΙΣΗ ΑΛΦΑΡΙΘΜΗΤΙΚΩΝ ΚΕΙΜΕΝΩΝ	65
14	ΠΟΛΥΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ	67
14.1	ΔΗΜΙΟΥΡΓΙΑ ΠΟΛΥΔΙΑΣΤΑΤΩΝ ΠΙΝΑΚΩΝ.....	67
14.2	ΔΗΜΙΟΥΡΓΙΑ ΠΟΛΥΔΙΑΣΤΑΤΩΝ ΠΙΝΑΚΩΝ ΜΕ ΤΗΝ ΒΟΗΘΕΙΑ ΣΥΝΑΡΤΗΣΕΩΝ	68
14.3	ΑΝΑΚΛΗΣΗ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ ΠΟΛΥΔΙΑΣΤΑΤΩΝ ΠΙΝΑΚΩΝ	68
14.4	ΠΟΛΥΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ ΚΕΙΜΕΝΟΥ	69
15	ΑΡΧΕΙΑ	71
15.1	ΆΝΟΙΓΜΑ ΚΑΙ ΚΛΕΙΣΙΜΟ ΑΡΧΕΙΩΝ	72
15.2	ΕΓΓΡΑΦΗ ΚΑΙ ΑΝΑΓΝΩΣΗ ΔΥΑΔΙΚΩΝ ΔΕΔΟΜΕΝΩΝ ΑΠΟ ΑΡΧΕΙΑ	74
15.3	ΕΓΓΡΑΦΗ ΚΑΙ ΑΝΑΓΝΩΣΗ ASCII ΔΕΔΟΜΕΝΩΝ ΑΠΟ ΑΡΧΕΙΑ	76
15.4	ΔΙΑΧΕΙΡΙΣΗ ΣΦΑΛΜΑΤΩΝ ΕΝΤΟΛΩΝ ΠΡΟΣΠΕΛΑΣΗΣ ΣΕ ΑΡΧΕΙΑ	79
15.5	ΜΕΤΑΚΙΝΗΣΗ ΤΟΥ ΔΕΙΚΤΗ ΠΡΟΣΠΕΛΑΣΗΣ ΑΡΧΕΙΟΥ	80
16	ΕΥΡΕΤΗΡΙΟ	84

1 ΤΙ ΕΙΝΑΙ Η MATLAB

Η MATLAB[®] είναι μια υψηλών επιδόσεων γλώσσα προγραμματισμού προσανατολισμένη στην επίλυση ειδικών προβλημάτων. Ενσωματώνει δυνατότητες αριθμητικών υπολογισμών, προγραμματισμού και γραφικών απεικονίσεων μέσα από ένα αρκετά εύχρηστο περιβάλλον όπου τα προβλήματα προς επίλυση και οι αντίστοιχες λύσεις τους απεικονίζονται με τρόπο που προσεγγίζουν τις αντίστοιχες μαθηματικές εκφράσεις. Χαρακτηριστικά αναφέρουμε μερικές από τις πιο συνηθισμένες χρήσεις της γλώσσας :

- ❖ Μαθηματικά και υπολογισμοί
- ❖ Ανάπτυξη αλγορίθμων
- ❖ Κατασκευή μοντέλων, προσομοιώσεις
- ❖ Ανάλυση δεδομένων, και απεικόνιση
- ❖ Γραφικές απεικονίσεις
- ❖ Ανάπτυξη εφαρμογών, καθώς και δημιουργία περιβάλλοντος εφαρμογής

Η MATLAB περιέχει ένα αλληλεπιδραστικό υπολογιστικό σύστημα στο οποίο η κύρια δομή δεδομένων είναι ο πίνακας. Η δυνατότητα αυτή μας επιτρέπει να κατασκευάσουμε κώδικα για την επίλυση πολλών τεχνικών υπολογιστικών προβλημάτων, με έμφαση σε εκείνα τα οποία εμπεριέχουν μετασχηματισμούς διανυσμάτων και πινάκων σε σχετικά σύντομο χρονικό διάστημα, συγκρινόμενο με τον χρόνο που απαιτείται για να αναπτύξουμε μία εφαρμογή σε γλώσσα που δεν περιέχει αλληλεπιδραστικό περιβάλλον όπως η C και η FORTRAN.

Το όνομα της γλώσσας προήλθε από τα αρχικά των λέξεων *Matrix Laboratory*. Ο κύριος σκοπός για τον οποίο δημιουργήθηκε ήταν για την όσον το δυνατόν ευκολότερη πρόσβαση σε λογισμικό αιχμής, όσον αφορά τις πράξεις μεταξύ πινάκων, που αναπτύχθηκαν κατά την διάρκεια προγραμμάτων όπως LINPACK και το EISPACK.

Η MATLAB εξελίχθηκε με την βοήθεια τω ίδιων των χρηστών της. Σε πανεπιστημιακά ιδρύματα αποτελεί το αναπόσπαστο εργαλείο για την διδασκαλία μαθηματικών, και τεχνικών επιστημών τόσο σε εισαγωγικό επίπεδο όσο και σε επίπεδο έρευνας. Στην βιομηχανία αποτελεί ιδανική επιλογή για ανάλυση, έρευνα και ανάπτυξη λογισμικού υψηλών απαιτήσεων.

Η MATLAB περιέχει λογισμικό ειδικού σκοπού (πακέτα) το οποίο επιτρέπει εκπαίδευση και εφαρμογή εξειδικευμένων τεχνολογιών με ελάχιστο κόστος. Το λογισμικό ειδικού σκοπού είναι συλλογές από αρχεία (M-Files) τα οποία περιέχουν συνήθως συναρτήσεις που επεκτείνουν τις δυνατότητες της MATLAB έτσι ώστε να μπορεί να επιλύσει ειδικά προβλήματα. Μερικά από τα πλέον διαδεδομένα πακέτα λογισμικού αφορούν συναρτήσεις Επεξεργασία σημάτων, Τεχνητής νοημοσύνης, Νευρωνικών Δικτύων, Wavelets, προσομοιώσεις γραμμικών συστημάτων κ.α.

2 ΤΟ ΣΥΣΤΗΜΑ MATLAB

Το σύστημα της MATLAB αποτελείται από τα εξής 5 κύρια μέρη :

1. Γλώσσα MATLAB. Είναι μια γλώσσα υψηλού επιπέδου που χρησιμοποιεί πίνακες, συναρτήσεις, δομές, έλεγχο ροής προγράμματος και διαθέτει χαρακτηριστικά αντικειμενοστραφούς προγραμματισμού. Τα επιμέρους χαρακτηριστικά της γλώσσας είναι οργανωμένα σε 6 καταλόγους (Directories), τους ακόλουθους:

ops	Τελεστές και ειδικοί χαρακτήρες
lang	Αρχές σχεδιασμού γλώσσας προγραμματισμού
strfun	Σειρές χαρακτήρων
iofun	Είσοδος έξοδος αρχείων
timefun	Ημερομηνία και ώρα
datatypes	Τύποι Δεδομένων και δομές

2. Το περιβάλλον εργασίας της MATLAB. Είναι ένα σύνολο εργαλείων που επιτρέπουν στον χρήστη να εργαστεί είτε σαν απλός χρήστης ο οποίος εκτελεί μεμονωμένες εντολές και συναρτήσεις είτε να εργαστεί σαν προγραμματιστής. Περιλαμβάνει δυνατότητες εισαγωγής-εξαγωγής δεδομένων από το περιβάλλον της, διαχείρισης, δημιουργίας και ανεύρεσης λαθών (Debugging) σε εφαρμογές της MATLAB. Οι εντολές και οι γενικές συναρτήσεις της MATLAB εμπεριέχονται στον κατάλογο:

general	Εντολές και συναρτήσεις γενικού σκοπού
---------	--

3. Διαχειριστής Γραφικών. Αποτελεί το γραφικό σύστημα της MATLAB. Περιλαμβάνει διαταγές για υψηλού επιπέδου 2-Διαστάσεων, 3-Διαστάσεων απεικονίσεις, επεξεργασία εικόνων, δυναμική κίνηση (animation), και παρουσίαση γραφικών. Δίνει επίσης την δυνατότητα χειρισμού εντολών χαμηλού επιπέδου για πλήρη προσαρμογή των γραφικών απεικονίσεων καθώς επίσης για την ανάπτυξη γραφικών εφαρμογών (GUI's). Οι συναρτήσεις γραφικών είναι διατεταγμένες σε 5 καταλόγους :

graph2d	Δισδιάστατες γραφικές παραστάσεις
graph3d	Τρισδιάστατες γραφικές παραστάσεις
specgraph	Εξειδικευμένες γραφικές απεικονίσεις
graphics	Διαχείριση γραφικών
uitools	Εργαλεία γραφικών εφαρμογών

4. Η Βιβλιοθήκη Μαθηματικών συναρτήσεων της MATLAB. Υπάρχει μια πλούσια συλλογή από αλγόριθμους υπολογισμού απλών συναρτήσεων, όπως οι αριθμητικές πράξεις πινάκων, τριγωνομετρικές συναρτήσεις όπως, ημίτονο, συνημίτονο, κοκ, καθώς και περισσότερο πολύπλοκες συναρτήσεις όπως η αντιστροφή πινάκων, συναρτήσεις Bessel, FFT κα. Η βιβλιοθήκη των συναρτήσεων είναι οργανωμένη σε 8 καταλόγους:

elmat	Βασικές συναρτήσεις διαχείρισης πινάκων
elfun	Βασικές μαθηματικές συναρτήσεις
specfun	Εξειδικευμένες μαθηματικές συναρτήσεις

matfun	Συναρτήσεις που αφορούν πίνακες και γραμμική άλγεβρα
datafun	Ανάλυση δεδομένων, FFT
polyfun	Πολυώνυμα και παρεμβολές
funfun	Επίλυση συνήθων διαφορικών εξισώσεων
sparfun	Διαχείριση αραιών πινάκων

5. Το περιβάλλον εφαρμογής προγράμματος API της MATLAB. Πρόκειται για μια βιβλιοθήκη συναρτήσεων που επιτρέπει στον χρήστη να κατασκευάσει εφαρμογές σε γλώσσα C και FORTRAN που αλληλεπιδρούν με την MATLAB. Περιλαμβάνει διευκολύνσεις για κλήση υπορουτίνων της MATLAB (Dynamic Linking), και για ανάγνωση και εγγραφή αρχείων με κατάληξη MAT (Mat-files).

2.1 Η εκκίνηση της MATLAB

Για να χρησιμοποιήσετε το αλληλεπιδραστικό περιβάλλον της MATLAB σε ένα PC, επιλέξτε με το ποντίκι το εικονίδιο της MATLAB. Για να τρέξετε την MATLAB σε ένα σύστημα που έχει λειτουργικό σύστημα UNIX δώστε στην γραμμή διαταγών `matlab`.

Για να βγείτε από την εφαρμογή δώστε `quit` στην γραμμή διαταγών. Επίσης μπορεί να βγείτε χρησιμοποιώντας την διαταγή `exit` από το μενού επιλογών.

2.2 Το παράθυρο εντολών

Το παράθυρο εντολών είναι το κύριο παράθυρο με το οποίο επικοινωνεί ο χρήστης με τον μεταγλωττιστή της MATLAB. Σε συστήματα UNIX, το παράθυρο εντολών είναι το τερματικό παράθυρο από το οποίο ξεκινάμε την MATLAB.

Το γραφικό περιβάλλον εργασίας απεικονίζει την προτροπή (`>>`) ενεργοποιώντας ταυτόχρονα τον αλληλεπιδραστικό μεταγλωττιστή και εκτελεστή εντολών και συναρτήσεων. Για παράδειγμα για να δώσουμε ένα 3x3 πίνακα πληκτρολογούμε :

```
A=[1 2 3 ; 4 5 6; 7 8 10]
```

Όταν δίνουμε **Enter** ή **Return** η MATLAB ανταποκρίνεται με

```
A=
 1  2  3
 4  5  6
 7  8 10
```

Για να αντιστρέψουμε τον πίνακα δίνουμε:

```
B=inv(A)
```

2.3 Διόρθωση εντολών στο παράθυρο εντολών

Υπάρχουν αρκετοί τρόποι χρησιμοποιώντας τα πλήκτρα Ctrl και τα τόξα του πληκτρολογίου (arrows) να κάνετε ανάκληση, διόρθωση και επαναχρησιμοποίηση διαταγών. Για παράδειγμα ας υποθέσουμε ότι δώσατε κατά λάθος:

```
rho=(1+sqrt(5))/2
```

Το λάθος εδώ εντοπίζεται στην συνάρτηση sqrt. Η σωστή σύνταξη της εντολής είναι sqrt. Η MATLAB ανταποκρίνεται με:

```
Undefined function or variable 'sqrt'.
```

Αντί να δώσουμε ξανά ολόκληρη την διαταγή, πατήστε το πλήκτρο \uparrow . Η λανθασμένη γραμμή εντολών απεικονίζεται ξανά στην οθόνη. Χρησιμοποιήστε τα πλήκτρα \leftarrow για να μετακινήσετε τον δρομέα και να εισάγετε το γράμμα r στην θέση που λείπει.

Επαναλαμβανόμενη χρήση του πλήκτρου \uparrow προκαλεί την ανάκληση προηγούμενων εντολών. Οι διαταγές που εισάγεται κατά την διάρκεια μίας εργασίας στην MATLAB αποθηκεύονται προσωρινά σε καταχωρητές της μνήμης. Μπορείτε να χρησιμοποιήσετε την έξυπνη ανάκληση (smart recall) για να καλέσετε προηγούμενες διαταγές στις οποίες υπάρχουν μερικοί αρχικοί χαρακτήρες τους οποίους καθορίζετε. Για παράδειγμα αν δώσετε τα γράμματα plo και πατήσετε \uparrow τότε μπορείτε να δείτε προηγούμενες διαταγές που άρχιζαν με το συνθετικό plo.

Ο πλήρης κατάλογος με τους συνδυασμούς των πλήκτρων που μπορείτε να χρησιμοποιήσετε δίνετε πιο κάτω:

\uparrow , ctrl-p	Ανάκληση προηγούμενης γραμμής.
\downarrow , ctrl-n	Κάλεσμα επόμενης γραμμής.
\leftarrow , ctrl-b	Μετακίνηση προς τα πίσω κατά ένα χαρακτήρα.
\rightarrow , ctrl-f	Μετακίνηση προς τα εμπρός κατά ένα χαρακτήρα.
Ctrl + \rightarrow , ctrl-r	Μετακίνηση προς τα δεξιά κατά μία λέξη.
Ctrl + \leftarrow , ctrl-l	Μετακίνηση προς τα αριστερά κατά μία λέξη.
Option - \rightarrow	Μετακίνηση προς τα δεξιά κατά μία λέξη.
Option - \leftarrow	Μετακίνηση προς τα αριστερά κατά μία λέξη.
Home, ctrl-a	Μετακίνηση στην αρχή της γραμμής.
End, ctrl-e	Μετακίνηση στο τέλος της γραμμής.
Esc, ctrl-u	Καθαρισμός γραμμής.
Del, ctrl-d	Διαγραφή χαρακτήρα στην θέση που βρίσκεται ο δρομέας.
Backspace, ctrl-h	Διαγραφή χαρακτήρα στην θέση πριν από τον δρομέα.
Ctrl-k	Διαγραφή γραμμής.

2.4 Η εντολή *format*

Η διαταγή `format` ελέγχει την αριθμητική μορφοποίηση των τιμών που απεικονίζονται στην οθόνη. Αυτή επηρεάζει μόνο την απεικόνιση των αριθμών και όχι την ακρίβεια εκτέλεσης των αριθμητικών υπολογισμών που εκτελεί η MATLAB. Πιο κάτω δίνουμε μερικές διαφορετικές μορφοποιήσεις:

```
x = [4/3 1.2345e-6]

format short

1.3333 0.0000

format short e

1.3333e+000 1.2345e-006

format short g

1.3333 1.2345e-006

format long

1.333333333333333 0.00000123450000

format long e

1.333333333333333e+000 1.234500000000000e-006

format long g

1.333333333333333 1.2345e-006

format bank

1.33 0.00

format rat

4/3 1/810045

format hex

3ff5555555555555 3eb4b6231abfd271
```

Εάν ένα στοιχείο του πίνακα είναι μεγαλύτερο από 10^3 ή μικρότερο από 10^{-3} η MATLAB εφαρμόζει ένα συντελεστή κλίμακας για `short` και `long` formats.

Επιπροσθέτως μπορείτε να χρησιμοποιήσετε διαταγές για μορφοποίηση της εξόδου όπως :

```
format compact
```

Η εντολή αυτή απαλείφει πολλές από τις κενές γραμμές που εμφανίζονται στην οθόνη. Αυτό μας επιτρέπει να βλέπουμε περισσότερη πληροφορία το παράθυρο της οθόνης που δουλεύουμε. Αν θέλετε περισσότερες δυνατότητες μορφοποίησης των δεδομένων που εκτυπώνονται χρησιμοποιήστε τις συναρτήσεις `fprint` και `sprintf`.

2.5 Καταστολή εξόδου

Αν δώσετε μια εντολή και πατήσετε το Enter ή το Return, η MATLAB αυτομάτως απεικονίζει το αποτέλεσμα στην οθόνη. Ωστόσο, αν στο τέλος της γραμμής δώσετε (;) η MATLAB πραγματοποιεί τους υπολογισμούς που απαιτούνται αλλά δεν παράγει στην οθόνη κανένα αποτέλεσμα. Αυτό είναι ιδιαίτερα χρήσιμο όταν δημιουργούμε μεγάλους πίνακες, όπως:

```
A = magic(100);
```

ή κατασκευάζουμε πρόγραμμα το οποίο αποτελείται από πολλές εντολές.

2.6 Μεγάλες γραμμές εντολών

Αν μια γραμμή διαταγής είναι αρκετά μεγάλη μπορείτε να χρησιμοποιήσετε τρεις (3) τελείες ακολουθούμενες από Enter ή Return για να δείξετε ότι η διαταγή συνεχίζει στην επόμενη γραμμή. Για παράδειγμα :

```
s = 1 - 1/2 + 1/3 - 1/4 + 1/5 - 1/6 + 1/7 ...
- 1/8 + 1/9 - 1/10 + 1/11 - 1/12;
```

Τα κενά στο +, -, = δεν είναι υποχρεωτικά αλλά βελτιώνουν πολύ την αναγνωσιμότητα του κώδικα και προφανώς συμβάλουν σημαντικά στην ευκολία εντοπισμού και διόρθωσης σφαλμάτων. Ο μέγιστος αριθμός χαρακτήρων που επιτρέπεται να τοποθετηθούν σε μια γραμμή εντολών της MATLAB είναι 4096.

3 Ο ΧΩΡΟΣ ΕΡΓΑΣΙΑΣ ΤΗΣ MATLAB

Ο χώρος εργασίας της MATLAB περιλαμβάνει ένα σύνολο μεταβλητών (arrays) που μπορείτε να χρησιμοποιήσετε από την γραμμή εκτέλεσης εντολών. Μπορείτε επίσης να χρησιμοποιήσετε τις διαταγές `who` και `whos` για να τις μεταβλητές που έχουν ήδη οριστεί στον χώρο εργασίας της MATLAB. Η διαταγή `who` δίνει έναν σύντομο κατάλογο, ενώ η εντολή `whos` δίνει επιπλέον πληροφορίες που αφορούν το μέγεθος και τον τύπο των μεταβλητών

Στο παράδειγμα που ακολουθεί δίνεται ένα παράδειγμα της εντολής `whos` για οκτώ μεταβλητές.

```
whos
```

Name	Size	Bytes	Class
A	4x4	128	double array
D	3x5	120	double array
M	10x1	40	cell array
S	1x3	628	struct array
h	1x11	22	char array
n	1x1	8	double array
s	1x5	10	char array
v	1x14	28	char array

Σύνολο 93 στοιχεία με χρησιμοποίηση 984 bytes

Για διαγραφή όλων των μεταβλητών από το χώρο εργασίας δώστε την εντολή `Clear`.

3.1 Άνοιγμα και αποθήκευση των μεταβλητών του χώρου εργασίας

Οι διαταγές της MATLAB `save` και `load` μας επιτρέπουν να αποθηκεύουμε τα περιεχόμενα του χώρου εργασίας σε κάθε χρονική στιγμή, και στην συνέχεια να επαναφέρουμε τα δεδομένα με την εκτέλεση μιας μόνο εντολής. Οι διαταγές `load` και `save` μας δίνουν επίσης την δυνατότητα να διαβάζουμε και να γράφουμε δεδομένα σε ASCII μορφή.

Η διαταγή `save` αποθηκεύει τα περιεχόμενα του χώρου εργασίας σε ένα MAT-file το οποίο περιέχει τις μεταβλητές αποθηκευμένες με δυαδικό τρόπο. Η εντολή:

```
save j182
```

αποθηκεύει όλες τις μεταβλητές του χώρου εργασίας στο αρχείο "j182.mat". Η επαναφορά των μεταβλητών στον χώρο εργασίας πραγματοποιείται με την εντολή `load`.

Αν επιθυμούμε μπορούμε να αποθηκεύσουμε μόνο ορισμένες μεταβλητές, πρέπει όμως

να καθορίσουμε τα ονόματα των μεταβλητών μετά το όνομα του αρχείου αποθήκευσης. Για παράδειγμα :

```
save je10 x y z
```

Η εντολή αποθηκεύει μόνο τις μεταβλητές *x*, *y*, και *z* στο αρχείο "je10.mat".

Σε συστήματα PC και Macintosh, η λειτουργία της αποθήκευσης είναι διαθέσιμη και από το μενού File.

3.2 Καθορισμός μορφοποίησης αρχείου (format)

Μπορούμε να επιλέξουμε τον τρόπο κωδικοποίησης των δεδομένων στα αρχεία αποθήκευσης χρησιμοποιώντας ορισμένα κλειδιά (flags) που τοποθετούμε στην εντολή αποθήκευσης:

-mat	Αποθήκευση δυαδικών δεδομένων (default)
-ascii	Χρησιμοποίηση 8-ψήφιων ASCII χαρακτήρων
-ascii -double	Χρησιμοποίηση 16-ψήφιων ASCII χαρακτήρων
-ascii -double -tabs	Οριοθέτηση στοιχείων με tabs
-v4	Αποθήκευση σε format της MATLAB έκδοση 4.
-append	Προσάρτηση στο τέλος του αρχείου (Append)

3.3 Διαβάζοντας ascii αρχεία

Μπορούμε να διαβάζουμε δεδομένα που είναι αποθηκευμένα σε αρχεία ASCII, ως εξής:

```
load tides.dat
```

Η εντολή δημιουργεί μια μεταβλητή με το όνομα *tides* στον χώρο εργασίας. Αν τα δεδομένα είναι διατεταγμένα σε *m* γραμμές με *n* τιμές σε κάθε μία από αυτές, τότε το αποτέλεσμα θα είναι να δημιουργηθεί ένας πίνακας διαστάσεων *m*×*n*.

3.4 Αποθήκευση δεδομένων όταν το όνομα αρχείου υπάρχει σε μεταβλητή χαρακτήρων

Αν το όνομα του αρχείου όπως και τα ονόματα των μεταβλητών είναι αποθηκευμένα σε μεταβλητή χαρακτήρων, μπορούμε να χρησιμοποιήσουμε τις αντίστοιχες συναρτήσεις *load* και *save* (αποθήκευση, ανάγνωση δυαδικών δεδομένων). Για παράδειγμα ο κώδικας:

```
save('myfile', 'VAR1', 'VAR2')
A = 'myfile';
load(A)
```

θα φέρει το ίδιο αποτέλεσμα με τις ακόλουθες δύο εντολές:

```
save myfile VAR1 VAR2
load myfile
```

Για αποθήκευση και ανάγνωση πολλαπλών αρχείων μπορούμε να χρησιμοποιήσουμε μια επαναληπτική διαδικασία. Για παράδειγμα, ο κώδικας που ακολουθεί αποθηκεύει τα τετράγωνα των αριθμών 1 έως 10 σε διαδοχικά αρχεία data1 έως και data10:

```
file = 'data';
for i = 1:10
    j = i.^2;
    save([file int2str(i)], 'j');
end
```

3.5 Δημιουργία αρχείων M-files

Ο PC Editor μας παρέχει την δυνατότητα εκτός από απλές λειτουργίες επεξεργασίας κειμένου να μπορούμε να έχουμε πρόσβαση στα εργαλεία εύρεσης λαθών (M-file Debugging Tools).

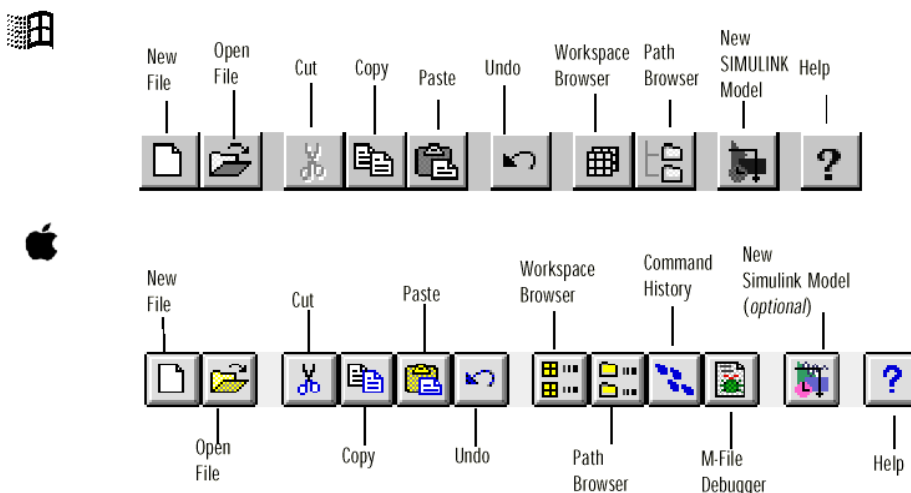
Ένας τρόπος για να ανοίξουμε ένα αρχείο M-file είναι χρησιμοποιώντας την προτροπή edit ακολουθούμενη από το όνομα του αρχείου π.χ:

```
edit poof
```

Ένας δεύτερος τρόπος είναι από τα μενού του Command Window.

- Για άνοιγμα ενός νέου αρχείου M-file, επιλέξτε New από το μενού File ή επιλέξτε το αντίστοιχο εικονίδιο της μπάρας
- Για άνοιγμα ενός υπάρχοντος αρχείου επιλέξτε Open από το μενού File ή επιλέξτε το εικονίδιο Open File.

Στο σχήμα που ακολουθεί απεικονίζονται τα εικονίδια του γραφικού περιβάλλον της MATLAB. Στην πρώτη γραμμή απεικονίζονται τα εικονίδια στο περιβάλλον των Windows της Microsoft, ενώ στην δεύτερη γραμμή απεικονίζονται τα εικονίδια στο περιβάλλον του λειτουργικού συστήματος Macintosh.



3.6 Διαδρομή αναζήτησης συναρτήσεων της MATLAB

Η MATLAB ακολουθεί μια μοναδική διαδρομή ανεύρεσης συναρτήσεων. Η διαδρομή αυτή ακολουθείται κάθε φορά που απαιτείται εντοπισμός αρχείων M-files. Τα αρχεία αυτά είναι ταξινομημένα σε διαφορετικούς καταλόγους στο σύστημά μας. Πολλά από αυτά προσφέρονται με την γλώσσα MATLAB ενώ άλλα παρέχονται ξεχωριστά με τα διάφορα πακέτα εφαρμογών (Toolboxes).

Αν εισάγετε το όνομα `fit` στην γραμμή προτροπής της MATLAB, ο μεταγλωττιστής της γλώσσας εκτελεί τις ακόλουθες ενέργειες με την σειρά που αυτές αναγράφονται:

1. Ελέγχει την ύπαρξη μεταβλητής `fit` στο πεδίο εργασίας
2. Ελέγχει για την ύπαρξη συνάρτησης με το όνομα `fit`
3. Ψάχνει στο χρησιμοποιούμενο κατάλογο για αρχείο με το όνομα `fit.m`
4. Ψάχνει σε όλους τους καταλόγους που είναι δηλωμένοι στην διαδρομή αναζήτησης για αρχεία με το όνομα `fit.m`

Αν έχετε περισσότερες από μία συναρτήσεις με το ίδιο όνομα στους καταλόγους που είναι δηλωμένοι στην διαδρομή αναζήτησης, μόνο η πρώτη που θα βρεθεί κατά την διαδικασία αναζήτησης θα εκτελεστεί. Οι υπόλοιπες δεν θα εκτελεστούν ποτέ, εκτός αν διαγραφεί το αρχείο που εκτελείται ή τροποποιηθεί το μονοπάτι αναζήτησης των εκτελέσιμων αρχείων.

3.7 Διαχείριση αρχείων και εντολές λειτουργικού συστήματος

Οι διαταγές `dir`, `type`, `delete`, και `cd` υλοποιούν ένα σύνολο από γενικές εντολές του λειτουργικού συστήματος. Ο πίνακας που ακολουθεί δίνει τις αντιστοιχίες βασικών εντολών για διαφορετικά λειτουργικά συστήματα :

MATLAB	MS-DOS	UNIX
<code>dir</code>	<code>dir</code>	<code>ls</code>
<code>type</code>	<code>type</code>	<code>cat</code>
<code>delete</code>	<code>del or erase</code>	<code>rm</code>
<code>cd</code>	<code>chdir</code>	<code>cd</code>

3.8 Εκτέλεση προγραμμάτων από το περιβάλλον εργασίας

Με την προσθήκη του χαρακτήρα `!` μπορούμε να εκτελέσουμε προγράμματα από το κέλυφος του περιβάλλοντος εργασίας της MATLAB. Η εκτέλεση αυτών των προγραμμάτων πραγματοποιείται στο κέλυφος του λειτουργικού συστήματος αλλά τα αποτελέσματα εκτυπώνονται στο παράθυρο του περιβάλλοντος εργασίας της MATLAB. Η λειτουργία αυτή είναι πολύ χρήσιμη διότι έχουμε την δυνατότητα να εκτελούμε άλλα προγράμματα ή εφαρμογές χωρίς να κλείσουμε το περιβάλλον εργασίας της MATLAB.

Για παράδειγμα σε λειτουργικό σύστημα UNIX η εντολή:

```
!vi darwin.m
```

προκαλεί την ενεργοποίηση του κειμενογράφου vi το οποίο με την σειρά του ανοίγει το αρχείο "darwin.m". Με το τέλος εκτέλεσης του προγράμματος, το λειτουργικό σύστημα αναθέτει πάλι τον έλεγχο του πληκτρολογίου στο περιβάλλον εργασίας της MATLAB.

3.9 Το αρχείο εκκίνησης m-file

Κατά την εκκίνηση της η MATLAB εκτελεί αυτόματα τις εντολές που υπάρχουν σε δύο αρχεία, το matlabrc.m, και αν υπάρχει, τις εντολές του αρχείου startup.m. Το αρχείο matlabrc.m, το οποίο βρίσκεται στον κατάλογο αρχείων που έχει εγκατασταθεί η MATLAB, είναι δεσμευμένο για χρήση από την εταιρεία κατασκευής του λογισμικού (MathWorks) και από τον διαχειριστή του λειτουργικού συστήματος. Το αρχείο startup.m μπορεί να τροποποιηθεί χρήστη. Στο αρχείο startup.m μπορείτε να τοποθετήσετε τμήματα κώδικα της MATLAB ο οποίος πρέπει να εκτελείται αυτόματα με την εκκίνηση του μεταγλωττιστή. Συνήθεις εντολές είναι τροποποίηση του καταλόγων αναζήτησης συναρτήσεων, ρυθμίσεις γραφικών κ.α.

Για παράδειγμα η εντολή που ακολουθεί προσθέτει ένα κατάλογο εργαλείων στην βασική διαδρομή αναζήτησης συναρτήσεων.

```
addpath /home/hss/myMatTools
```

4 ΜΕΤΑΒΛΗΤΕΣ ΚΑΙ ΣΤΑΘΕΡΕΣ

Υπάρχουν έξι βασικοί τύποι δεδομένων στην MATLAB, ο καθένας εκ των οποίων μπορεί να είναι και πολυδιάστατος πίνακας. Αυτοί οι τύποι είναι `double`, `char`, `sparse`, `uint8`, `cell`, `struct`. Στον πίνακα που ακολουθεί φαίνονται αναλυτικά οι τύποι των μεταβλητών:

<i>Τύπος</i>	<i>Παράδειγμα</i>	<i>Περιγραφή</i>
<code>double</code>	<code>[1 2 ; 3 4]</code> <code>5+6i</code>	Πίνακας διπλής ακρίβειας (αυτός είναι ο πιο συνηθισμένος τύπος στην MATLAB).
<code>char</code>	<code>'Hello'</code>	Πίνακας χαρακτήρων (κάθε χαρακτήρας έχει μέγεθος 16 bit). Αναφέρεται και ως <code>string</code> .
<code>sparse</code>	<code>Speye(5)</code>	Αραιός πίνακας διπλής ακρίβειας (διδιάστατος μόνο).
<code>cell</code>	<code>{17 'hello' eye(2)}</code>	Πίνακας κελιών. Κάθε κελί αυτού του πίνακα περιέχει ένα άλλο πίνακα. Πίνακας δομών.
<code>struct</code>	<code>a.day = 12 ;</code> <code>a.color = 'Red'</code> <code>a.mat = magic(3) ;</code>	
<code>uint8</code>	<code>Uint8(magic(3))</code>	Πίνακας με στοιχεία τα οποία είναι 8 bit ακέραιοι χωρίς πρόσημο αριθμοί από 0 ως 255 (Δεν γίνονται πράξεις μεταξύ αυτών των αριθμών).
<code>userobject</code>	<code>Inline('sin(x)')</code>	Στοιχεία ορισμένα από τους χρήστες.

4.1 Ειδικές μεταβλητές και σταθερές

Πολλές συναρτήσεις επιστρέφουν μερικές ειδικές τιμές οι οποίες μπορούν να χρησιμοποιηθούν στα M-Files. Μαζί με τις σταθερές που είναι ήδη ορισμένες έχουμε τον ακόλουθο πίνακα:

<code>ans</code>	Το πιο πρόσφατο αποτέλεσμα κάποιας πράξης
<code>eps</code>	Η ακρίβεια κινητής υποδιαστολής
<code>realmax</code>	Ο μεγαλύτερος αριθμός κινητής υποδιαστολής
<code>realmin</code>	Ο μικρότερος αριθμός κινητής υποδιαστολής
<code>pi</code>	3.1415926...
<code>i, j</code>	Φανταστική μονάδα
<code>inf</code>	Άπειρο
<code>NaN</code>	Δείκτης αδυναμίας εκτέλεσης αριθμητικής πράξης
<code>computer</code>	Ο τύπος του υπολογιστή
<code>flops</code>	Μέτρηση των υπολογισμών κινητής υποδιαστολής
<code>version</code>	Έκδοση της MATLAB

4.2 Δεικτοδότηση και τμήματα πινάκων

Το στοιχείο της σειράς i και στήλης j ενός πίνακα A ορίζεται ως $A(i,j)$. Το $A(4,2)$ είναι το στοιχείο του πίνακα που βρίσκεται στην τέταρτη γραμμή και στην δεύτερη στήλη. Επίσης μπορούμε να αναφερθούμε σε στοιχεία κάποιου πίνακα με ένα μόνο δείκτη π.χ. $A(I)$ στην περίπτωση κατά την οποία έχουμε διανύσματα γραμμών ή στηλών.

Όταν προσπελάσουμε ένα στοιχείο ενός πίνακα το οποίο βρίσκεται εκτός των ορίων του, τότε η MATLAB επιστρέφει το ακόλουθο μήνυμα λάθους:

```
Index exceeds matrix dimensions
```

Αντίθετα, όταν κάποιος προσπαθήσει να αποθηκεύσει μια τιμή σε ένα στοιχείο εκτός των ορίων του πίνακα, τότε το μέγεθος του αυξάνει αυτόματα για να χωρέσει το καινούργιο στοιχείο.

Όταν θέλουμε να αναφερθούμε σε τμήματα ενός πίνακα χρησιμοποιούμε εκφράσεις που περιέχουν άνω-κάτω τελείες (:). Στην έκφραση που ακολουθεί αναφερόμαστε στα πρώτα k στοιχεία της j -στήλης του πίνακα A

```
A(1:k , j)
```

Έτσι για τον πιο πάνω πίνακα A το άθροισμα των στοιχείων της τέταρτης στήλης δίνεται από την εντολή:

```
sum(A(1:4 , 4))
```

ή ισοδύναμα από την εντολή:

```
sum(A(:, end))
```

Έτσι λοιπόν γίνεται φανερό ότι οι άνω-κάτω τελεία αναφέρεται σε *όλα* τα στοιχεία μιας γραμμής ή μιας στήλης ενός πίνακα, ενώ η εντολή `end` αναφέρεται στην τελευταία γραμμή ή στήλη του πίνακα.

4.3 Ενοποίηση πινάκων

Η διαδικασία ενοποίησης πινάκων (Concatenation) έχει στόχο τον σχηματισμό ενός μεγαλύτερου πίνακα ο οποίος περιέχει την αριθμητική τιμή μικρότερων πινάκων. Για την ενοποίηση πινάκων χρησιμοποιούμε τις αγκύλες (`[]`). Αν θεωρήσουμε τον πίνακα A του προηγούμενου παραδείγματος, τότε ο νέος πίνακας B :

```
B = [A A+32; A+48 A+16]
```

```
B =
16 3 2 13 48 35 34 45
5 10 11 8 37 42 43 40
9 6 7 12 41 38 39 44
4 15 14 1 36 47 46 33
64 51 50 61 32 19 18 29
```

```
53 58 59 56 21 26 23 28
57 54 55 60 25 22 23 28
52 63 62 49 20 31 30 17
```

Για να διαγράψουμε γραμμές ή στήλες ενός πίνακα χρησιμοποιούμε πάλι τις διπλές αγκύλες (`[]`). Συνεπώς για την διαγραφή της δεύτερης στήλης του πίνακα A θα δόσουμε την εντολή:

```
A(:, 2) = [] ;
```

```
A =
16 2 13
5 11 8
9 7 12
4 14 1
```

Προσοχή πρέπει να δοθεί όσον αφορά στη διαγραφή γραμμών ή στηλών ενός πίνακα. Δεν μπορεί να διαγραφεί κάποιο στοιχείο όταν το αποτέλεσμα της διαγραφής δεν είναι πίνακας. Το ίδιο βέβαια ισχύει και κατά την ενοποίηση πινάκων. Για τον πίνακα του προηγούμενου παραδείγματος η έκφραση

```
A(1, 2) = []
```

θα έχει ως αποτέλεσμα την εμφάνιση μηνύματος λάθους.

5 ΤΕΛΕΣΤΕΣ ΠΡΑΞΕΩΝ

Οι τελεστές πράξεων στην MATLAB διακρίνονται σε 3 κατηγορίες:

- Αριθμητικοί οι οποίοι εκτελούν πράξεις όπως το άθροισμα, το γινόμενο αριθμών και πινάκων κλπ.
- Τελεστές σχέσεων οι οποίοι συγκρίνουν σταθερές ή μεταβλητές όπως μεγαλύτερο, μικρότερο, ίσο, άνισο κλπ.
- Οι λογικοί τελεστές (AND, OR, NOT) έχουν δυαδικό αποτέλεσμα: TRUE ή FALSE.

Από τους πιο πάνω τελεστές μεγαλύτερη προτεραιότητα έχουν οι αριθμητικοί και ακολουθούν οι τελεστές σχέσεις και οι λογικοί.

5.1 Αριθμητικοί Τελεστές

Οι κανόνες προτεραιότητας για τους αριθμητικούς τελεστές είναι:

1. Αναστροφή (.'), ύψωση σε δύναμη (^), αναστροφή συζυγούς μιγαδικού πίνακα ('), ύψωση σε δύναμη ενός πίνακα (^).
2. Άθροιση (+), αφαίρεση (-).
3. Πολλαπλασιασμός στοιχείων (*), δεξιά διαίρεση (./), αριστερή διαίρεση (\), πολλαπλασιασμός πινάκων (*), αριστερή διαίρεση πινάκων (\), δεξιά διαίρεση πινάκων (/).
4. Πρόσθεση (+), αφαίρεση (-).
5. Τελεστής στήλης (:).

Να σημειωθεί ότι η προτεραιότητα των τελεστών μπορεί να παρακαμφθεί με την χρήση παρενθέσεων. Σε αυτή την περίπτωση οι τελεστές που βρίσκονται μέσα στις παρενθέσεις εκτελούνται πρώτες.

5.2 Τελεστές Σχέσεων

Υπάρχουν έξι είδη τελεστών σχέσης:

<	Μικρότερο από
<=	Μικρότερο ίσο από
>	Μεγαλύτερο από
>=	Μεγαλύτερο ίσο από
==	Ίσο με
~=	Όχι ίσο με

Η σύγκριση με την βοήθεια των τελεστών σχέσης γίνεται μεταξύ στοιχείων πινάκων με ίδιες διαστάσεις. Για διανύσματα και τετραγωνικούς πίνακες και τα δύο ορίσματα πρέπει να έχουν το ίδιο μέγεθος εκτός και αν το ένα είναι βαθμωτό. Σε αυτή την περίπτωση η MATLAB ελέγχει ένα προς ένα το βαθμωτό μέγεθος με κάθε στοιχείο του

πίνακα. Στα σημεία που ο τελεστής είναι αληθής η MATLAB επιστρέφει 1 (TRUE) ενώ σε όσα στοιχεία δεν ικανοποιείται η συνθήκη του τελεστή σχέσης τοποθετείται η τιμή 0 (FALSE). Οι τελεστές αυτού του είδους χρησιμοποιούνται κυρίως με τις εντολές `if`, `for`, `while`, `switch`.

Οι τελεστές σχέσης εφαρμόζονται πάντοτε σε στοιχεία πινάκων και διανυσμάτων, όπως στο παράδειγμα που ακολουθεί:

```
A =[2 7 6 ;9 0 -1 ;3 0.5 6] ;
B = [8 0.2 0; -3 2 5 ; 4 -1 7] ;
A < B
Ans =
     1     0     0
     0     1     1
     1     0     1
```

5.3 Λογικοί Τελεστές

Οι λογικοί τελεστές που χρησιμοποιούνται στην MATLAB είναι οι AND, OR και NOT:

```
&      AND
|      OR
~      NOT
```

Η κατασκευή συνθετότερων λογικών εκφράσεων μπορεί να πραγματοποιηθεί με την βοήθεια των τελεστών σχέσης. Οι λογικοί τελεστές μπορούν να πραγματοποιήσουν λογικές πράξεις και μεταξύ στοιχείων πινάκων με ίδιες διαστάσεις. Για διανύσματα και τετραγωνικούς πίνακες και τα δύο ορίσματα πρέπει να έχουν το ίδιο μέγεθος εκτός και αν το ένα είναι βαθμωτό. Σε αυτή την περίπτωση η MATLAB ελέγχει ένα προς ένα το βαθμωτό μέγεθος με κάθε στοιχείο του πίνακα. Σε κάθε στοιχείο του πίνακα ο τελεστής επιστρέφει 1 (TRUE) ή 0 (FALSE) ανάλογα με το αποτέλεσμα της λογικής πράξης.

6 ΠΙΝΑΚΕΣ ΚΑΙ ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ

Ο πίνακας στην MATLAB είναι ένα σύνολο πραγματικών ή μιγαδικών αριθμών. Όλες οι αριθμητικές πράξεις που ορίζονται από την γραμμική άλγεβρα υποστηρίζονται από την MATLAB. Οι συναρτήσεις γραμμικής άλγεβρας εντοπίζονται στο κατάλογο `matfun`.

Κατηγορία Συνάρτηση Περιγραφή

Ανάλυση Πινάκων-----

<code>norm</code>	Matrix or vector norm.
<code>normest</code>	Estimate the matrix 2-norm.
<code>rank</code>	Matrix rank.
<code>det</code>	Determinant.
<code>trace</code>	Sum of diagonal elements.
<code>Null</code>	Null space.
<code>orth</code>	Orthogonalization.
<code>Rref</code>	Reduced row echelon form.
<code>Subspace</code>	Angle between two subspaces

Γραμμικές εξισώσεις και επίλυση συστήματος γραμμικών εξισώσεων-----

<code>inv</code>	Matrix inverse.
<code>cond</code>	Condition number for inversion.
<code>condest</code>	1-norm condition number estimate.
<code>chol</code>	Cholesky factorization.
<code>cholinc</code>	Incomplete Cholesky factorization.
<code>lu</code>	LU factorization.
<code>luinc</code>	Incomplete LU factorization.
<code>qr</code>	Orthogonal-triangular decomposition.
<code>npls</code>	Nonnegative least-squares.
<code>pinv</code>	Pseudoinverse.
<code>lscov</code>	Least squares with known covariance.

Ιδιοτιμές πινάκων-----

<code>eig</code>	Eigenvalues and eigenvectors.
<code>svd</code>	Singular value decomposition.
<code>eigs</code>	A few eigenvalues.
<code>svds</code>	A few singular values.
<code>poly</code>	Characteristic polynomial.
<code>polyeig</code>	Polynomial eigenvalue problem.
<code>condeig</code>	Condition number for eigenvalues.
<code>hess</code>	Hessenberg form.
<code>qz</code>	QZ factorization.
<code>schur</code>	Schur decomposition.

Συναρτήσεις πινάκων-----

<code>expm</code>	Matrix exponential.
<code>logm</code>	Matrix logarithm.
<code>sqrtm</code>	Matrix square root.
<code>funm</code>	Evaluate general matrix function.

6.1 Πίνακες στην MATLAB

Τυπικά οι όροι μήτρα και πίνακας χρησιμοποιούνται στην βιβλιογραφία. Για την ακρίβεια μήτρα, είναι ένας δισδιάστατος τετραγωνικός πίνακας με πραγματικές ή φανταστικές τιμές. Οι γραμμικές αλγεβρικές πράξεις μεταξύ πινάκων βρίσκουν εφαρμογή σε μια πλειάδα επιστημονικών πεδίων. Η MATLAB έχει πάρα πολλές συναρτήσεις που δημιουργούν διαφορετικά είδη πινάκων. Δύο από αυτές θα χρησιμοποιηθούν για την δημιουργία δύο πινάκων 3x3 οι οποίες θα χρησιμοποιηθούν στα παραδείγματα που ακολουθούν. Το πρώτο παράδειγμα αναφέρεται σε έναν συμμετρικό πίνακα.

```
A = pascal(3)
```

```
A =
     1  1  1
     1  2  3
     1  3  6
```

Η δεύτερος πίνακας είναι μη συμμετρικός:

```
B = magic(3)
```

```
B =
     8  1  6
     3  5  7
     4  9  2
```

Ένα άλλο παράδειγμα είναι ένας πίνακας 3x2.

```
C = fix(10*rand(3,2))
```

```
C =
     9  4
     2  8
     6  7
```

Ένα διάνυσμα στήλης είναι ένας πίνακας mx1, ενώ ένα διάνυσμα γραμμής είναι ένας πίνακας 1xn. Επίσης στοιχείο ενός πίνακα είναι πάλι ένας πίνακας διαστάσεων 1x1. Οι παρακάτω δηλώσεις :

```
u = [3; 1; 4]
```

```
v = [2 0 -1]
```

```
s = 7
```

παράγουν ένα διάνυσμα στήλης, ένα διάνυσμα γραμμής και έναν πίνακα στοιχείο

```
u =
```

```
3
```

```
1
```

```
4
```

```
v =
```

```
2 0 -1
```

```
s =
```

```
7
```

6.2 Πρόσθεση και αφαίρεση πινάκων

Η πρόσθεση και η αφαίρεση πινάκων ορίζονται ακριβώς όπως αυτές των μαθηματικών πινάκων.

Προσθέτοντας A και B και μετά αφαιρώντας το A δίνει το B.

```
X = A + B
```

```
X =
```

```
9 2 7
```

```
4 7 10
```

```
5 12 8
```

```
Y = X - A
```

```
Y =
```

```
8 1 6
```

```
3 5 7
```

```
4 9 2
```

Η πρόσθεση και η αφαίρεση πινάκων απαιτούν την ύπαρξη πινάκων με τις ίδιες διαστάσεις, ή κάποιος όρος να είναι μήτρα στοιχείο.

Αν οι διαστάσεις των πινάκων δεν είναι ίδιες, τότε έχουμε την πρόκληση μηνύματος λάθους:

```
X = A + C
```

```
Error using ==> +
Matrix dimensions must agree.
```

```
w = v + s
```

```
w =
```

9 7 6

6.3 Γινόμενο πινάκων και ανάστροφος πίνακας

Ένα διάνυσμα στήλη και ένα διάνυσμα γραμμή μπορούν να πολλαπλασιαστούν. Το αποτέλεσμα μπορεί να είναι ένα στοιχείο, το εσωτερικό ή το εξωτερικό γινόμενο.

$$x = v * u$$

$$x = 2$$

$$X = u * v$$

$$X = \begin{bmatrix} 6 & 0 & -3 \\ 2 & 0 & -1 \\ 8 & 0 & -4 \end{bmatrix}$$

Για πίνακες πραγματικών αριθμών η διαδικασία της αναστροφής μετασχηματίζει τα a_{ij} σε a_{ji} . Η MATLAB χρησιμοποιεί μία απόστροφο για να δηλώσει την αναστροφή ενός πίνακα. Στο παράδειγμά μας ο A είναι συμμετρικός έτσι ο A' είναι ίσος με τον A. Αλλά για τον B που δεν είναι συμμετρικός έχω:

$$X = B'$$

$$X = \begin{bmatrix} 8 & 3 & 4 \\ 1 & 5 & 9 \\ 6 & 7 & 2 \end{bmatrix}$$

Η αναστροφή μετασχηματίζει ένα διάνυσμα γραμμή σε διάνυσμα στήλη.

$$x = v'$$

$$x = \begin{bmatrix} 2 \\ 0 \\ -1 \end{bmatrix}$$

Αν τώρα οι x, y είναι διανύσματα στήλης πραγματικών αριθμών, το γινόμενο $x * y$ δεν ορίζεται αλλά τα δύο γινόμενα :

$$x' * y$$

και

$$y' * x$$

δίνουν το ίδιο πίνακα στοιχείο.

Για ένα διάνυσμα ή μήτρα φανταστικών τιμών z, η ποσότητα z' , δεικνύει τον ανάστροφο συζυγή μιγαδικό. Έτσι αν :

$$z = [1+2i \ 3+4i]$$

τότε z' είναι

$$\begin{matrix} 1-2i \\ 3-4i \end{matrix}$$

ενώ $z.'$ είναι

$$\begin{matrix} 1+2i \\ 3+4i \end{matrix}$$

Για μιγαδικά διανύσματα, τα γινόμενα $x' * y$ και $y' * x$ είναι συζυγή μεταξύ τους ενώ το γινόμενο $x' * x$ ενός φανταστικού διανύσματος είναι διάνυσμα πραγματικό.

Το γινόμενο δύο πινάκων $C=AB$ ορίζεται όταν ο πίνακας A έχει διαστάσεις $m \times p$ και ο B $p \times n$, τότε ο C έχει διαστάσεις $m \times n$. Ο πολλαπλασιασμός των πινάκων μπορεί στην MATLAB να γίνει με την βοήθεια των επαναληπτικών βρόχων :

```
for i = 1:m
    for j = 1:n
        C(i,j) = A(i,:)*B(:,j);
    end
end
```

Η MATLAB χρησιμοποιεί το σύμβολο $*$ για να δηλώσει τον πολλαπλασιασμό πινάκων. Ισοδύναμα ο πολλαπλασιασμός πινάκων γράφεται

$$C = A * B ;$$

Τα παρακάτω παραδείγματα δεικνύουν το γεγονός ότι μεταξύ των πινάκων γενικά η αντιμεταθετικότητα του πολλαπλασιασμού αυτών δεν ισχύει :

$$X = A*B$$

$$X = \begin{matrix} 15 & 15 & 15 \\ 26 & 38 & 26 \\ 41 & 70 & 39 \end{matrix}$$

$$Y = B*A$$

$$Y = \begin{matrix} 15 & 28 & 47 \\ 15 & 34 & 60 \\ 15 & 28 & 43 \end{matrix}$$

Ένας πίνακας μπορεί να πολλαπλασιαστεί από τα αριστερά με ένα διάνυσμα στήλη ενώ από τα δεξιά με ένα διάνυσμα γραμμή.

```

x = A*u
x =
     8
    17
    30

y = v*B
y =
    12  -7  10

```

Οι διαστάσεις των πινάκων θα πρέπει να ικανοποιούν την συνθήκη: ο αριθμός γραμμών του πρώτου πίνακα πρέπει να είναι ίσος με τον αριθμό στηλών του δεύτερου προκειμένου να μπορεί να γίνει ο πολλαπλασιασμός αυτών:

```

X = A*C
X =
    17  19
    31  41
    51  70

Y = C*A

Error using ==> *
Inner matrix dimensions must agree.

```

Οποιοσδήποτε πίνακας μπορεί να πολλαπλασιαστεί με μία μήτρα στοιχείο

```

w = s*v
w =
    14  0  -7

```

6.4 Μοναδιαίος πίνακας

Ο μαθηματικός συμβολισμός για τον μοναδιαίο πίνακα είναι I . Ο πίνακας αυτός έχει το χαρακτηριστικό ότι τα διαγώνια στοιχεία του είναι 1 ενώ όλα τα υπόλοιπα είναι μηδενικά.

Επίσης ισχύει ότι $AI = A$ και $IA = A$. Η συνάρτηση `eye(m,n)` της MATLAB επιστρέφει έναν $m \times n$ μοναδιαίο πίνακα ενώ η `eye(n)` επιστρέφει έναν $n \times n$ μοναδιαίο πίνακα.

6.5 Το γινόμενο kronecker tensor

Το γινόμενο Kronecker, $\text{Kron}(X,Y)$, δύο πινάκων είναι ένας πίνακας όλων των δυνατών γινομένων των στοιχείων του X με αυτά του Y . Αν ο X είναι διαστάσεων $m \times n$ ενώ ο Y

είναι διαστάσεων pxq τότε ο $\text{Kron}(X,Y)$ είναι διαστάσεων $mrxnq$. Τα στοιχεία διατάσσονται με την εξής σειρά :

$$\begin{bmatrix} X(1,1)*Y & X(1,2)*Y & \dots & X(1,n)*Y \\ \vdots & \vdots & \vdots & \vdots \\ X(m,1)*Y & X(m,2)*Y & \dots & X(m,n)*Y \end{bmatrix}$$

Η διαδικασία αυτή χρησιμοποιείται συχνά για τον επανασηματισμό μικρών πινάκων. Για παράδειγμα :

$$X = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

και $I = \text{eye}(2,2)$ τότε οι πίνακες Kronecker είναι

`kron(X,I)`

$$\begin{bmatrix} 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 2 \\ 3 & 0 & 4 & 0 \\ 0 & 3 & 0 & 4 \end{bmatrix}$$

`kron(I,X)`

$$\begin{bmatrix} 1 & 2 & 0 & 0 \\ 3 & 4 & 0 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 3 & 4 \end{bmatrix}$$

6.6 Μέτρο διανυσμάτων και πινάκων

Το μέτρο τάξης p του διανύσματος x ,

$$\|x\|_p = \left(\sum x_i^p\right)^{1/p}$$

υπολογίζεται από την συνάρτηση `norm(x,p)`. Το μέτρο ορίζεται για τιμές $p > 1$ αλλά οι πιο συχνές τιμές του p είναι 1,2 και ∞ . Η προεπιλεγμένη τιμή του p είναι 2 η οποία αντιστοιχεί στην Ευκλείδεια απόσταση.

`[norm(v,1) norm(v) norm(v,inf)]`

$$\text{ans} = \begin{bmatrix} 3.0000 & 2.2361 & 2.0000 \end{bmatrix}$$

Το μέτρο τάξης p του πίνακα A ορίζεται να είναι

$$\|A\|_p = \max_{\mathbf{x}} \frac{\|A\mathbf{x}\|_p}{\|\mathbf{x}\|_p}$$

υπολογίζεται από την $\text{norm}(A, p)$. Η προεπιλεγμένη τιμή του p είναι 2.

```
[norm(C,1) norm(C) norm(C,inf)]
ans =
    19.0000    14.8015    13.0000
```

6.7 Επίλυση γραμμικών εξισώσεων

Ένα από τα πιο σημαντικά προβλήματα που συναντώνται στον χώρο της επίλυσης τεχνικών προβλημάτων είναι το εξής: δοθέντος δύο πινάκων A και B υπάρχει μοναδικός πίνακας X έτσι ώστε

$$AX=B$$

ή

$$XA=B$$

Η MATLAB χρησιμοποιεί τους συμβολισμούς \backslash και $\backslash\backslash$ για να δηλώσει αντιστοίχως τις λύσεις των εξισώσεων :

$X = A \backslash B$ δεικνύει την λύση της εξίσωσης $AX = B$.
 $X = B / A$ δεικνύει την λύση της εξίσωσης $XA = B$.

Επίσης ισχύει

$$(B/A)' = (A' \backslash B')$$

Ο πίνακας A δεν απαιτείται να είναι τετραγωνικός.

Στην γενική περίπτωση, όταν ο A είναι πίνακας με διαστάσεις $m \times n$ υπάρχουν τρεις περιπτώσεις :

$m = n$.	Εύρεση μιας και μοναδικής λύσης.
$m > n$.	Εύρεση λύσης με την μέθοδο ελαχίστων τετραγώνων.
$m < n$.	Εύρεση λύσης με τουλάχιστον m μη μηδενικές λύσεις.

6.7.1 Εύρεση μοναδικής λύσης

Το πλέον συχνά εμφανιζόμενο πρόβλημα είναι η περίπτωση κατά την οποία ο πίνακας A είναι τετραγωνικός και ο πίνακας B να είναι ένα διάνυσμα στήλης. Έχουμε δηλαδή ένα σύστημα n γραμμικών εξισώσεων με n αγνώστους. Είναι γνωστό ότι η λύση είναι $x=A^{-1}b$:

```
x = A \ u
x =
    10
   -12
```

Εύκολα μπορεί να επιβεβαιωθεί ότι ο $A*x$ έχει τις ίδιες διαστάσεις με το u .
Αν οι πίνακες A και B είναι τετραγωνικοί τότε $X=A/B$:

```
X = A\B
X =
    19    -3   -1
   -17     4    13
     6     0   -6
```

6.7.2 Αντίστροφοι πίνακες και ορίζουσες

Εάν ο A είναι τετραγωνικός πίνακας, οι εξισώσεις $AX=I$ και $XA=I$, έχουν την ίδια λύση X . Η λύση X καλείται αντίστροφος του A και συμβολίζεται με A^{-1} και υπολογίζεται από την συνάρτηση `inv`. Η συνάρτηση `det` υπολογίζει την ορίζουσα ενός τετραγωνικού πίνακα :

```
d = det(A)
X = inv(A)

d =
    1

X =
     3    -3     1
    -3     5    -2
     0    -2     1
```

Επειδή ο A είναι συμμετρικός, έχει ακέραιες τιμές για στοιχεία, και η ορίζουσά του είναι 1, το ίδιο θα ισχύει και για τον αντίστρόφο του.

```
d = det(B)
X = inv(B)

d =
   -360

X =
    0.1472 -0.1444 0.0639
   -0.0611 0.0222 0.1056
   -0.0194 0.1889 -0.1028
```

6.7.3 Ψευδοαντίστροφοι πίνακες

Οι μη τετραγωνικοί πίνακες δεν έχουν αντίστροφους ή ορίζουσες. Τουλάχιστον μία από τις εξισώσεις $AX=I$ και $XA=I$ δεν έχει λύση. Μια μερική αντικατάσταση του αντιστρόφου πίνακα είναι ο Moore-Penrose ψευδοαντίστροφος, ο οποίος υπολογίζεται από την συνάρτηση `pinv`.

```
X = pinv(C)
X =
    0.1159 -0.0729 0.0171
   -0.0534 0.1152 0.0418
```

Ο πίνακας

```
Q = X*C
Q =
    1.0000 0.0000
    0.0  1.0000
```

είναι ο 2x2 μοναδιαίος αλλά πίνακας

```
P = C*X
P =
    0.8293 -0.1958 0.3213
   -0.1958 0.7754 0.3685
    0.3213 0.3685 0.3952
```

δεν είναι ο 3x3 μοναδιαίος. Αν ο A είναι mxn, m>n και full rank n τότε κάθε μία από τις τρεις δηλώσεις :

```
x = A\b
x = pinv(A)*b
x = inv(A'*A)*A'*b
```

θεωρητικά υπολογίζει την ίδια λύση X, μόνο που ο τελεστής \ κάνει την διαδικασία γρηγορότερη.

7 ΕΛΕΓΧΟΣ ΡΟΗΣ

Υπάρχουν τέσσερις εντολές για έλεγχο ροής εκτέλεσης του προγράμματος στην MATLAB:

- Η εντολή `if`, μαζί με τα `else`, `elseif`, εκτελεί ένα τμήμα του προγράμματος όταν η λογική έκφραση της εντολής είναι αληθής.
- Η εντολή `switch`, μαζί με `case`, `otherwise`, εκτελεί ένα τμήμα του προγράμματος ανάλογα με το περιεχόμενο μιας έκφρασης ή μιας μεταβλητής.
- Η εντολή `while` εκτελεί ένα κομμάτι του προγράμματος όσο λογική συνθήκη που δίνεται στην εντολή είναι αληθής.
- Η εντολή `for` εκτελεί ένα κομμάτι του προγράμματος για έναν προκαθορισμένο αριθμό επαναλήψεων.

7.1 Εντολές ελέγχου ροής προγράμματος

if, else, and elseif

Η εντολή `if` υπολογίζει την τιμή της λογική έκφραση και εκτελεί ένα σύνολο από εντολές το Πρόγραμμα όταν η λογική έκφραση είναι αληθής. Η σύνταξη της εντολής είναι:

```
if λογική_έκφραση
    Πρόγραμμα
end
```

Αν η λογική έκφραση είναι αληθής (1), τότε η MATLAB εκτελεί το πρόγραμμα που βρίσκεται μεταξύ του `if` και του `end`. Αν η έκφραση είναι ψευδής (0), τότε η εκτέλεση του προγράμματος συνεχίζει μετά από την εντολή `end`.

```
if rem(a,2) == 0
    disp('a is even')
    b = a/2 ;
end
```

Αν η λογική έκφραση αντιστοιχεί σε κάποιο μη βαθμωτό μέγεθος τότε η έκφραση είναι αληθής όταν όλα τα στοιχεία του πίνακα έχουν μη μηδενική τιμή.

Για παράδειγμα έστω ότι το `X` είναι πίνακας . Τότε η έκφραση:

```
if X
    πρόγραμμα
end
```

είναι ισοδύναμη με την:

```
if all(X(:))
    πρόγραμμα
end
```

Με την εντολή `else` και `elseif` μπορούμε να κατασκευάσουμε πολυπλοκότερες εκφράσεις:

- Η εντολή `else` δεν εισάγει επιπρόσθετη λογική συνθήκη. Το πρόγραμμα που ακολουθεί την `else` εκτελείται μόνο αν η λογική συνθήκη του `if` που αντιστοιχεί στην `else` είναι ψευδής.
- Το `elseif` προσθέτει μία επιπλέον λογική συνθήκη η οποία ελέγχεται μόνο αν η λογική συνθήκη του προηγούμενου `if` είναι ψευδής (0). Το πρόγραμμα που ακολουθεί το `elseif` εκτελείται μόνο όταν η αντίστοιχη λογική έκφραση είναι αληθής. Μπορούν να υπάρχουν πολλαπλά `elseif` μέσα σε ένα `if`.

Στο παράδειγμα που ακολουθεί το πρόγραμμα-1 εκτελείται όταν είναι αληθής η λογική_έκφραση1. Όταν η λογική_έκφραση1 είναι ψευδής και η λογική_έκφραση2 είναι αληθής εκτελείται το πρόγραμμα-2. Όταν η λογική_έκφραση1 και η λογική_έκφραση2 είναι ψευδής εκτελείται το πρόγραμμα-3.

```
if λογική_έκφραση1
    Πρόγραμμα-1
elseif λογική_έκφραση2
    Πρόγραμμα-2
else
    Πρόγραμμα-3
end
```

switch

Η εντολή `switch` εκτελεί ένα σύνολο εντολών ανάλογα με το περιεχόμενο μιας μεταβλητής ή μιας έκφρασης. Η βασική μορφή της εντολής είναι:

Η MATLAB ελέγχει το περιεχόμενο της μεταβλητής ή υπολογίζει την τιμή της έκφραση. Εκτελείται το πρόγραμμα1 όταν το αποτέλεσμα των υπολογισμών δώσει την τιμή1, εκτελείται το πρόγραμμα2 όταν το αποτέλεσμα των υπολογισμών δώσει την τιμή2, κ.ο.κ. Αν το αποτέλεσμα των υπολογισμών δεν δώσει κάποια τιμή από τις σταθερές που αναγράφονται στις δηλώσεις `case` τότε εκτελείτε το πρόγραμμα που υπάρχει κάτω από την δήλωση `otherwise` (ΠρόγραμμαN). Αν δεν έχει τοποθετηθεί η δήλωση `otherwise` δεν εκτελείται κανένα από τα προγράμματα που βρίσκονται μέσα στην `switch`.

```
switch μεταβλητή ή έκφραση
    case τιμή1
        Πρόγραμμα1
    case τιμή2
        Πρόγραμμα2
    .
```



```

    .
    .
    otherwise
        ΠρόγραμμαN
end

```

while

Ο βρόγχος `while` εκτελεί το πρόγραμμα επαναληπτικά μέχρι όπου η έκφραση να γίνει ψευδής. Η σύνταξη της εντολής είναι:

```

while έκφραση
    Πρόγραμμα
end

```

Αν η έκφραση αναφέρεται σε όνομα πίνακα, τότε πρέπει όλα τα στοιχεία του να είναι αληθή (μη-μηδενικές τιμές) για να εκτελεστεί το πρόγραμμα.

Αν θέλετε να χρησιμοποιήσετε πολύπλοκες εκφράσεις πινάκων μπορείτε να χρησιμοποιήσετε τις εντολές `all` και `any`.

Στο παράδειγμα που ακολουθεί με την βοήθεια της `while` βρίσκουμε τον μικρότερο ακέραιο n για τον οποίο το $n!$ περιέχει περισσότερα από 100 ψηφία:

```

n = 1 ;
while prod(1:n) < 1e100
    n = n + 1 ;
end

```

for

Ο βρόχος `for` εκτελεί τμήμα του προγράμματος επαναληπτικά για ένα συγκεκριμένο αριθμό επαναλήψεων. Η σύνταξη της εντολής είναι:

```

for δείκτης = αρχή:βήμα:τέλος
    Πρόγραμμα
end

```

Αν δεν δοθεί το βήμα τότε αυτό θεωρείται ότι έχει την αριθμητική τιμή 1. Ως βήμα μπορεί να θεωρηθεί οποιοσδήποτε ακέραιος αριθμός ακόμη και αν αυτός είναι αρνητικός. Όταν το βήμα είναι θετικός αριθμός τότε ο βρόχος τερματίζεται όταν η τιμή του δείκτη γίνει μεγαλύτερη από την τιμή `τέλος`. Όταν αντίστοιχα το βήμα είναι αρνητικός αριθμός, τότε ο βρόχος τερματίζεται όταν ο δείκτης γίνει μικρότερος από την τιμή `τέλος`. Για παράδειγμα ο πιο κάτω βρόχος εκτελείται 5 φορές:

```

for i =2:6
    x(i) = 2*x(i-1) ;
end

```

Μπορούν να χρησιμοποιηθούν και ένθετοι (nested) βρόχοι με εντολές `for`:

```

for i = 2:m
    for j = 1:n
        A(i,j) = 1/(i+j-1) ;
    end
end

```

Ο δείκτης σε ένα βρόχο `for` μπορεί να είναι πίνακας. Σε κάθε επανάληψη του βρόχου ο δείκτης είναι ένα διάνυσμα που περιέχει μία στήλη του πίνακα, αρχίζοντας από την πρώτη και με αυξανόμενα βήματα να λάβει σαν τελική τιμή την τελευταία στήλη του πίνακα. Για παράδειγμα ας θεωρήσουμε τον `m×n` πίνακα `A`. Η έκφραση:

```

for i = A
    Πρόγραμμα
end

```

θέτει αρχικά την μεταβλητή `i` ίση με την πρώτη στήλη του πίνακα `A(:,1)`. Για την δεύτερη επανάληψη του βρόχου η μεταβλητή του δείκτη λαμβάνει σαν τιμή το περιεχόμενο της δεύτερης στήλης του πίνακα `kok`.

break

Ο τερματισμός οποιουδήποτε βρόχου πραγματοποιείται με την εντολή `break`.

8 ΒΟΗΘΕΙΑ ΚΑΙ ΤΕΚΜΗΡΙΩΣΗ

Υπάρχουν αρκετοί τρόποι πρόσβασης για πληροφορίες που αφορούν λειτουργίες της MATLAB, άμεσης βοήθειας και ερμηνείας των εντολών και των συναρτήσεων της:

1. Η διαταγή help
2. Το παράθυρο βοήθειας help
3. Η διαταγή lookfor
4. Το Help Desk της MATLAB
5. Η εκτύπωση σελίδων αναφοράς
6. Το Web Site της MathWorks

8.1 Η διαταγή help

Η διαταγή help είναι ο πιο κοινός τρόπος για την εξακρίβωση της σύνταξης και της συμπεριφοράς κάθε συνάρτησης. Η πληροφορία αποτυπώνεται κατευθείαν στην οθόνη του τερματικού, είτε στο παράθυρο του χώρου εργασίας στην περίπτωση που χρησιμοποιούμε γραφικό περιβάλλον. Η εντολή:

```
help magic
```

απεικονίζει

```
MAGIC Magic square.
MAGIC(N) is an N-by-N matrix constructed from
the integers 1 through N^2 with equal row,
column, and diagonal sums.
Produces valid magic squares
```

Όλες οι συναρτήσεις της MATLAB είναι οργανωμένες σε λογικά σύνολα. Για παράδειγμα όλες οι συναρτήσεις της γραμμικής Άλγεβρας βρίσκονται στον κατάλογο matfun. Για να δούμε τα ονόματα όλων των συναρτήσεων που βρίσκονται σε ένα κατάλογο, με πλήρη περιγραφή αυτών :

```
help matfun
Matrix functions - numerical linear algebra.
Matrix analysis.
norm - Matrix or vector norm.
normest - Estimate the matrix 2-norm
```

...

Η διαταγή

```
help
```

παράγει στην οθόνη έναν κατάλογο με την περιγραφή διαφόρων κατηγοριών συναρτήσεων:

```
matlab/general
matlab/ops
...
```

8.2 Παράθυρο help

Το παράθυρο βοήθειας είναι διαθέσιμο σε συστήματα PC και Macintoshes με την επιλογή την εντολής Help Window που βρίσκεται στο μενού επιλογών Help, ή επιλέγοντας το εικονίδιο με το ερωτηματικό που βρίσκεται στο κάτω μέρος της μπάρας εργασίας. Επίσης μπορείτε να έχετε πρόσβαση σε αυτό πληκτρολογώντας στην γραμμή εντολών:

```
helpwin
```

Για την χρησιμοποίηση του παραθύρου βοήθειας σε ένα συγκεκριμένο θέμα δώστε :

```
helpwin θέμα
```

Το παράθυρο βοήθειας σας δίνει τις ίδιες πληροφορίες όπως και η διαταγή help, αλλά η παραθυρική απεικόνιση σας επιτρέπει γρήγορες διασυνδέσεις σε άλλα θέματα.

8.3 Η διαταγή lookfor

Η διαταγή αυτή σας επιτρέπει την εύρεση συναρτήσεων και λειτουργιών βασισμένη σε μία λέξη κλειδί. Ψάχνει όλο το κείμενο βοήθειας για κάθε συνάρτηση της MATLAB που συσχετίζεται με την λέξη κλειδί, και επιστρέφει τις γραμμές του κειμένου βοήθειας που περιέχουν την λέξη κλειδί.

Για παράδειγμα η MATLAB δεν έχει καμία συνάρτηση που να ονομάζεται inverse. Έτσι η απάντηση της MATLAB στην εντολή :

```
help inverse
```

είναι

```
inverse.m not found.
```

Αν όμως δώσουμε την εντολή:

```
lookfor inverse
```

η MATLAB βρίσκει ένα αρκετά μεγάλο πλήθος συναρτήσεων που ενδεχομένως να συσχετίζονται με την λέξη κλειδί που έχουμε δώσει. Ο κατάλογος των συναρτήσεων σχετίζεται άμεσα με τα Toolboxes που έχετε εγκαταστήσει. Τυπικό παράδειγμα καταλόγου συσχετιζόμενων συναρτήσεων για την λέξη κλειδί inverse είναι:

```
INVHILB Inverse Hilbert matrix.
ACOSH Inverse hyperbolic cosine.
```

ERFINV Inverse of the error function.
INV Matrix inverse.
PINV Pseudoinverse.
IFFT Inverse discrete Fourier transform.
IFFT2 Two-dimensional inverse discrete Fourier transform.
ICCEPS Inverse complex cepstrum.
IDCT Inverse discrete cosine transform.

9 Ο ΔΙΟΡΘΩΤΗΣ ΣΦΑΛΜΑΤΩΝ ΤΗΣ MATLAB

Ο διορθωτής σφαλμάτων (Debugger) της MATLAB μας βοηθάει στην εύρεση σφαλμάτων στους κώδικες που αναπτύσσουμε σε MATLAB. Ο διορθωτής σφαλμάτων έχει μια γραμμή προτροπής η οποία είναι διαθέσιμη σε όλα τα λειτουργικά συστήματα. Σε λειτουργικά συστήματα που διαθέτουν γραφικό περιβάλλον υπάρχει και η αντίστοιχη έκδοση του γραφικού περιβάλλοντος.

9.1 Διαδικασία διόρθωσης σφαλμάτων

Η διόρθωση σφαλμάτων είναι μια διαδικασία που μας επιτρέπει να απομονώνουμε και να διορθώνουμε λάθη μέσα στον κώδικα που έχουμε δημιουργήσει. Η διαδικασία αυτή μας βοηθάει να διορθώνουμε δύο ειδών λάθη :

- Συντακτικά λάθη, όπως η κατά λάθος ανορθόγραφη εισαγωγή συναρτήσεων, ή η παράλειψη παρενθέσεων.
- Λάθη εκτέλεσης. Αυτά τα λάθη είναι κυρίως αλγοριθμικής φύσεως.

Μπορούμε να διορθώσουμε εύκολα τα συντακτικά λάθη βασιζόμενοι στα μηνύματα που μας απεικονίζει η MATLAB. Τα λάθη εκτέλεσης είναι σαφώς πιο δύσκολο να εντοπιστούν και γι' αυτό χρησιμοποιούμε κάποια από τις παρακάτω τεχνικές :

- Απομακρύνουμε τις δηλώσεις των (;) από τα αρχεία M-file. Με την απομάκρυνση αυτή εξαναγκάζουμε την MATLAB να απεικονίζει τα αποτελέσματα στην οθόνη.
- Τοποθετήστε Keyboard statements στα αρχεία M-file. Η λειτουργία αυτή δηλώνεται από την προτροπή "K>>".
- Κάντε σχόλιο την κύρια συνάρτηση και τρέξτε το M-file σαν Script.
- Χρησιμοποιήστε τον διορθωτή σφαλμάτων της MATLAB.

9.2 Ένα παράδειγμα διόρθωσης σφάλματος

Για να δοκιμάσετε τον διορθωτή σφαλμάτων δημιουργήστε ένα M-file που ονομάζεται `variance.m` το οποίο δέχεται ένα διάνυσμα εισόδου και επιστρέφει την εκτίμηση της διασποράς των τιμών του διανύσματος. Επίσης δημιουργήστε ένα άλλο αρχείο M-file `sqsum`, που υπολογίζει την μέση αφαιρούμενη τετραγωνική άθροιση από το διάνυσμα εισόδου

```
function y = variance(x)
mu = sum(x)/length(x);
tot = sqsum(x,mu);
y = tot/(length(x)-1);
```

Δημιουργήστε το `sqsum.m` file όπως ακριβώς φαίνεται στις γραμμές που ακολουθούν με ένα εσκεμμένο λάθος:

```
function tot = sqsum(x,mu)
tot = 0;
```

```

for i = 1:length(mu)
tot = tot + ((x(i)-mu).^2);
end

```

9.2.1 Εκτέλεση του προγράμματος

Δοκιμάστε τα M-files να δείτε αν λειτουργούν σωστά. Χρησιμοποιήστε την συνάρτησης της MATLAB std για τον υπολογισμό των αποτελεσμάτων :

Δημιουργήστε δύο διανύσματα εισόδου στην γραμμή προτροπής :

```
v = [1 2 3 4 5];
```

Υπολογίστε την διασπορά για κάθε ένα χρησιμοποιώντας την συνάρτηση std:

```

var1 = std(v).^2
var1 =
2.5000

```

Τώρα δοκιμάστε με την συνάρτηση της διασποράς :

```

myvar1 = variance(v)
myvar1 =
1

```

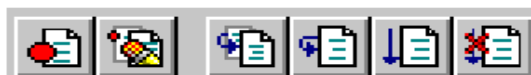
Η απάντηση είναι λανθασμένη. Ας χρησιμοποιήσουμε τον διορθωτή σφαλμάτων για να απομονώσουμε το λάθος στα M-files.







9.2.2 Διόρθωση σφαλμάτων σε Windows95/98/NT

Για να ξεκινήσετε την διαδικασία διόρθωσης σφαλμάτων εκτελέστε τις ακόλουθες ενέργειες:

- Εάν έχετε μόλις δημιουργήσει τα αρχεία M-files με τον Editor/Debugger μπορείτε να συνεχίσετε στο επόμενο βήμα
- Εάν έχετε δημιουργήσει τα αρχεία M-files με την βοήθεια κάποιου εξωτερικού Editor επιλέξτε το Open M-file στο Παράθυρο εκτέλεσης εντολών.

Ο Pc Editor/Debugger περιλαμβάνει ένα πλήθος εικονιδίων των οποίων η λειτουργία ερμηνεύεται στον πίνακα που ακολουθεί.



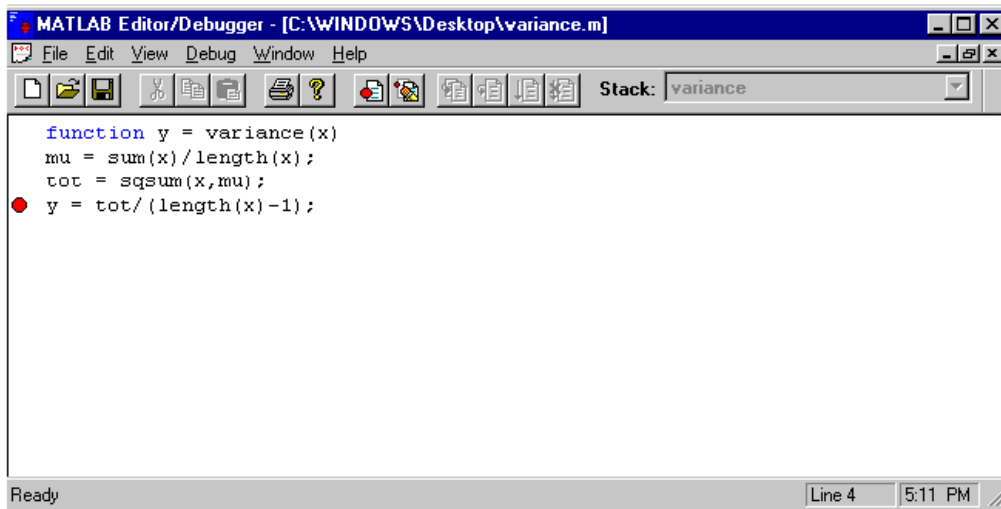
Toolbar Button	Description	Equivalent Command
	Set/Clear Breakpoint; set or clear a breakpoint at the line containing the cursor.	dbstop/ dbclear
	Clear All Breakpoints; clear all breakpoints that are currently set.	dbclear all
	Step In; execute the current line of the M-file and if the line is a call to another function, step into that function.	dbstep in
	Single Step; execute the current line of the M-file.	dbstep
	Continue; continue execution of M-file until completion or until another breakpoint is encountered.	dbcont
	Quit Debugging; exit the debugging state.	dbquit

9.2.3 Τοποθέτηση σημείων διακοπής εκτέλεσης προγράμματος

Οι περισσότερες εργασίες με τον διορθωτή σφαλμάτων ξεκινούν με τον καθορισμό ενός αριθμού διακοπών εκτέλεσης του προγράμματος (breakpoints). Οι διακοπές εκτέλεσης σταματούν την εκτέλεση του προγράμματος το οποίο βρίσκεται μέσα σε M-file σε συγκεκριμένες γραμμές έτσι ώστε να μας δώσουν την δυνατότητα να εκτυπώσουμε ή να αλλάξουμε τις τιμές των μεταβλητών και των πινάκων του προγράμματος. Ένα κόκκινο σημάδι stop δίπλα σε μια γραμμή δείχνει ότι σε αυτή την γραμμή έχουμε τοποθετήσει μία διακοπή εκτέλεσης προγράμματος.

Κατά την έναρξη της διαδικασίας διόρθωσης προγράμματος δεν γνωρίζουμε που βρίσκεται το λάθος στην συνάρτηση της διασποράς ή ακόμα αν αυτό βρίσκεται στο αρχείο variance.m ή το αρχείο squm.m. Μια λογική τοποθεσία για να βάλουμε ένα breakpoint είναι μετά από το $y = \text{tot}/(\text{length}(x)-1)$. Ανοίγουμε το αρχείο vm.m και τοποθετούμε μία διακοπή εκτέλεσης προγράμματος στην τέταρτη γραμμή.

Ο αριθμός γραμμής απεικονίζεται στο κάτω μέρος του παραθύρου. Τοποθετήστε το σημείο διακοπής προγράμματος με την βοήθεια του κουμπιού Breakpoint icon. Επίσης μπορείτε να επιλέξετε Set Breakpoint από το μενού Debug ή να κάνετε δεξιό κλικ για να φέρετε το αναδυόμενο μενού και να επιλέξετε Set/Clear Breakpoint.

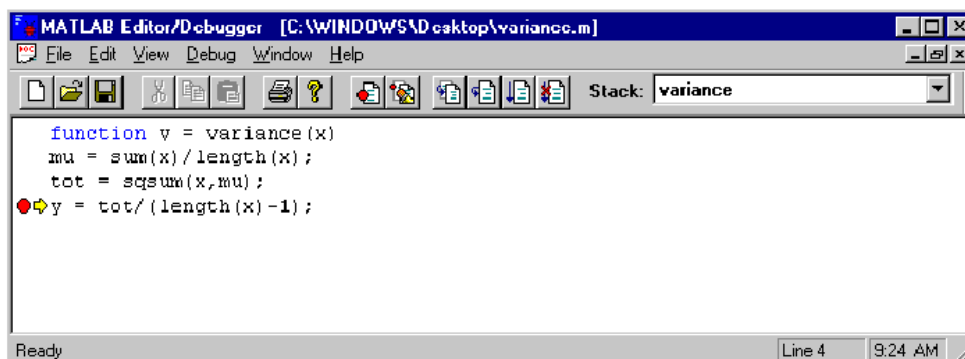


9.2.4 Έλεγχος μεταβλητών

Για να φτάσουμε στο σημείο που έχουμε τοποθετήσει το διακοπή εκτέλεσης προγράμματος, πρώτα θα πρέπει να εκτελέσουμε την συνάρτηση από το παράθυρο εκτέλεσης εντολών. Γι'αυτό τον λόγο εκτελούμε την εντολή:

```
variance(v)
```

Όταν η εκτέλεση ενός M-file σταματήσει στο σημείο που έχουμε τοποθετήσει την διακοπή εκτέλεσης προγράμματος, ένα κίτρινο βέλος προς τα δεξιά δείχνει την επόμενη προς εκτέλεση γραμμή. Ενώ ένα βέλος προς τα κάτω στο αριστερό μέρος του κειμένου δείχνει ότι θα σταματήσουμε στο τέλος της συνάρτησης ή του script. Αυτό μας επιτρέπει να εξετάζουμε τις μεταβλητές πριν την επιστροφή στην καλούσα συνάρτηση.



Ελέγξτε τις τιμές των `mu` και `tot` από τον διορθωτή σφαλμάτων. Επιλέξτε την κάθε μεταβλητή και επιλέγοντας το δεξί πλήκτρο του ποντικιού δίνουμε Evaluate Selection. Επίσης μπορείτε να επιλέξετε το Evaluate selection από το μενού View.

Η επιλογή και τα αποτελέσματα απεικονίζονται στην οθόνη στο παράθυρο εκτέλεσης εντολών:

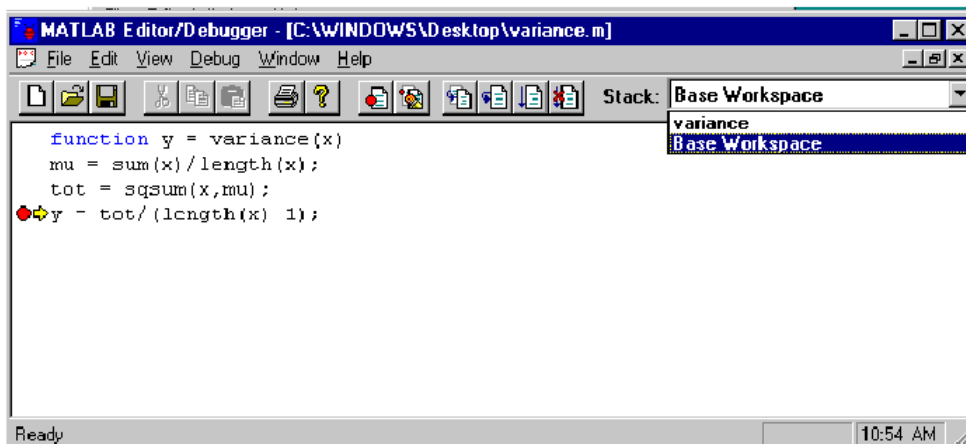
```
mu =
    3
tot =
    4
```

Το πρόβλημα εντοπίζεται στην συνάρτηση `sqsum`:

```
K mu »
K tot »
```

9.2.5 Αλλαγή του χώρου εργασίας-Τερματισμός εκτέλεσης συνάρτησης

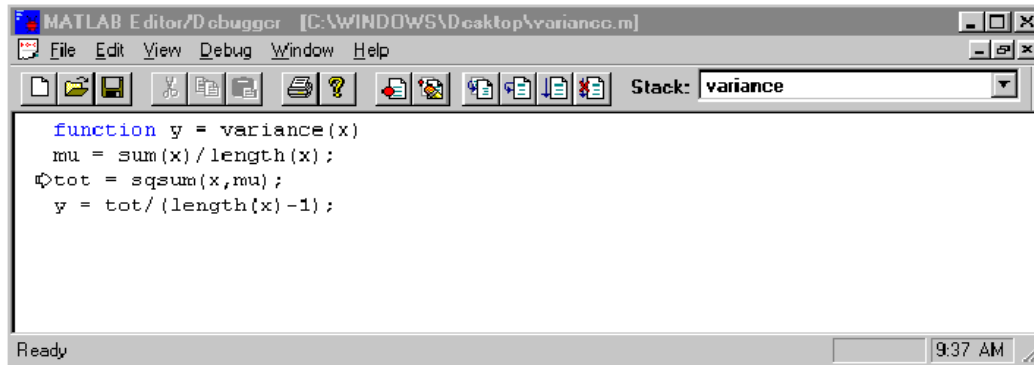
Χρησιμοποιήστε το μενού Stack για να αλλάξετε τα περιεχόμενα του workspace. Για να βγείτε από την συνάρτηση `variance` και να δείτε όλο το workspace, επιλέξτε Base.



9.2.6 Συνέχιση εκτέλεσης του προγράμματος

Αποσύρετε το σημείο διακοπής προγράμματος από την τέταρτη γραμμή του αρχείου `variance.m` τοποθετώντας τον δρομέα στην γραμμή και επιλέγοντας Clear Breakpoint από το μενού Debug. Συνεχίστε την εκτέλεση επιλέγοντας Continue από το μενού Debug.

Ανοίξτε το αρχείο `sqsum.m` και τοποθετείστε ένα σημείο διακοπής προγράμματος στην τέταρτη γραμμή του προγράμματος για να ελέγξετε τους υπολογισμούς που γίνονται στο επαναληπτικό μέρος του κώδικα. Τρέξτε το `variance` και πάλι χρησιμοποιώντας το ίδιο διάνυσμα εισόδου. Η εκτέλεση σταματά στην γραμμή 4 του `sqsum`. Αν κοιτάξουμε στην συνάρτηση `variance` από τον Editor/Debugger θα δούμε ότι η κλήση της `sqsum` δηλώνεται με ένα διαγραμμισμένο βέλος :



Ελέγξτε τον δείκτη i του βρόχου :

```
i =
     1
Κ i >
```

Στην συνέχεια επιλέξτε Single Step από το μενού Debug και εκτελέστε την επόμενη γραμμή.

Τυπώστε το περιεχόμενο της μεταβλητής tot :

```
tot =
     4
Κ tot >
```

Επιλέξτε και πάλι Single Step. Η sqsum διέρχεται από τον βρόχο μόνο μία φορά:

```
for i = 1:length(mu)
```

Ο βρόχος εκτελείται μέχρι length(mu). Στην συνέχεια επιλέξτε Quit Debugging από το μενού Debug για να σταματήσετε την εκτέλεση του M-file.

Για να δείτε αν αλλάζοντας την μεταβλητή tot έχουμε το σωστό αποτέλεσμα, δημιουργήστε ένα νέο σημείο διακοπής προγράμματος στην γραμμή 4 του variance.m και εκτελέστε πάλι το variance

```
variance(v)
```

Η εκτέλεση του προγράμματος σταματά μετά την επιστροφή του ελέγχου της sqsum, αλλά ακριβώς πριν η variance χρησιμοποιήσει την επιστρεφόμενη τιμή tot. Από το παράθυρο εκτέλεσης εντολών, δώστε στην μεταβλητή tot την τιμή 10.

```
Κ >> tot = 10

tot =
    10
```

Επιλέξτε Continue Execution από το Debug μενού.

9.2.7 Τερματισμός διαδικασίας διόρθωσης σφαλμάτων

Επιλέξτε Exit Editor/Debugger από το μενού File για τερματισμό της διαδικασίας debugging.

9.3 Διόρθωση προγραμμάτων από την γραμμή εκτέλεσης εντολών

Υπάρχουν μερικές διαταγές που πραγματοποιούν Debug για M-Files από την γραμμή προτροπής. Μια περιγραφή των πιο συνηθισμένων εντολών φαίνονται στον παρακάτω πίνακα :

Description	Syntax
Set breakpoint.	dbstop at line_num in file_name
Remove breakpoint.	dbclear at line_num in file_name
Stop on warning error or NaN/Inf Generation	dstop if warning error naninf infnan
Resume execution.	dbcont
List function call stack.	dbstack
List all breakpoints.	dbstatus file_name
Execute one or more lines.	dbstep nlines
List M-file with line numbers.	dbtype file_name
Change local workspace context (down).	dbdown
Change local workspace context (up).	dbup
Quit debug mode.	dbquit

9.4 Παράδειγμα διόρθωσης προγραμμάτων από το παράθυρο εκτέλεσης εντολών

Μπορείτε να πραγματοποιήσετε όλες τις επιμέρους λειτουργίες Debugging από την γραμμή εντολών. Για να μπορέσετε να κατανοήσετε το παρακάτω παράδειγμα χρησιμοποιήσετε τα M-files που έχουμε ήδη δημιουργήσει :

```
function y = variance(x)
mu = sum(x)/length(x);
tot= sqsum(x,mu);
y = tot/(length(x)-1);

function tot = sqsum(x,mu)
tot = 0;
for i = 1:length(mu)
tot = tot + ((x(i)-mu).^2);
end
```

9.4.1 Καθορισμός σημείων διακοπής προγράμματος

Η εντολή dbstop καθορίζει ένα σημείο διακοπής προγράμματος. Το αρχείο M-file σταματά πριν την εκτέλεση της γραμμής που έχει σημείο διακοπής προγράμματος.

```
dbstop variance 3
dbstop variance 4
```

9.4.2 Μετακίνηση μέσα στο κώδικα

Θεωρείστε ως είσοδο το διάνυσμα $v = [1 \ 2 \ 3 \ 4 \ 5]$. Οι αναμενόμενες τιμές για το `mean` και το `mean-removed` άθροισμα είναι αντίστοιχα 3 και 10. Ας δούμε αν τα αποτελέσματα που περιμένουμε στα σημεία των διακοπών προγράμματος είναι αυτές που πρέπει.

Αρχικά εκτελέστε την συνάρτηση από την γραμμή διαταγών :

```
variance(v)
```

Η MATLAB απεικονίζει την επόμενη προς εκτέλεση γραμμή, και τον αριθμό της :

```
3 tot = sqsum(x,mu);
K>>
```

Όταν η εκτέλεση σταματά σε ένα σημείο διακοπής προγράμματος, είσαστε αυτόματα στο Keyboard mode, όπως δηλώνεται εξάλλου από την ένδειξη `K>>`. Από το σημείο αυτό μπορείτε να δώσετε οποιαδήποτε διαταγή της MATLAB. Όταν τελειώσετε δώστε `dbcont` για να συνεχίσετε την εκτέλεση του προγράμματος από το σημείο που έχετε σταματήσει.

Όταν η εκτέλεση σταματά στο πρώτο σημείο διακοπής προγράμματος, χρησιμοποιήστε την εντολή `whos` για να δείτε τις διαθέσιμες μεταβλητές:

```
whos
```

Για να δείτε την τιμή της `mu` δώστε:

```
K>> mu
mu =
    3
```

Στην συνέχεια χρησιμοποιήστε την συνάρτηση `dbstep` για εκτέλεση μιας γραμμής. Όταν η εκτέλεση σταματήσει και πάλι πριν την γραμμή 4, ελέγξτε την τιμή της μεταβλητής `tot` για να δείτε αν αυτή έχει την αναμενόμενη τιμή :

```
K>> dbstep
4 y = tot/(length(x)-1);

K>> tot
tot =
    4
```

Φαίνεται λοιπόν πάλι ότι υπάρχει πρόβλημα στην συνάρτηση `sqsum`.

9.4.3 Αλλαγή του χώρου εργασίας-Τερματισμός εκτέλεσης συνάρτησης

Χρησιμοποιήστε τις συναρτήσεις `dbup` και `dbdown` για να μετακινηθείτε μεταξύ των χώρων εργασίας των συναρτήσεων και του βασικού χώρου εργασίας. Για να πάτε στο βασικό χώρο εργασίας από τον χώρο εργασίας της συνάρτησης `variance` δώστε :

```
dbup
whos
```

Η μεταβλητή `v` που χρησιμοποιούμε για έλεγχο καθώς και οι υπόλοιπες μεταβλητές φαίνονται στον κατάλογο που τυπώνεται. Για να επανέλθουμε στο χώρο εργασίας της `variance` δώστε :

```
dbdown
```

9.4.4 Εμφάνιση m-file με αριθμούς γραμμών

Δώστε `dbtype` για να δείτε την `sqsum`. Ορίστε σημεία διακοπής προγράμματος για να ελέγξετε του βρόχους επανάληψης και τους υπολογισμούς που γίνονται στο εσωτερικό των συναρτήσεων:

```
K>> dbtype sqsum

1 function tot = sqsum(x,mu)
2 tot = 0;
3 for i = 1:length(mu)
4 tot = tot + ((x(i)-mu).^2);
5 end

K>> dbstop sqsum 4
K>> dbstop sqsum 5
```

9.4.5 Απεικόνιση του καταλόγου εκτελούμενων συναρτήσεων

Επανέλθετε από το Keyboard mode δίνοντας `dbquit`. Στην γραμμή εντολών της MATLAB δώστε :

```
dbclear variance
```

για απομάκρυνση όλων των σημείων διακοπής προγράμματος από την συνάρτηση `variance`, ενώ διατηρείτε ανέπαφα τα σημεία διακοπής της συνάρτησης `sqsum`.

Εκτελέστε πάλι την συνάρτηση `variance` χρησιμοποιώντας το ίδιο διάνυσμα εισόδου:

```
variance(v)
4 tot = tot + ((x(i)-mu).^2);
```

Χρησιμοποιήστε την `dbstack` για να δείτε την λίστα με τις κλήσεις της συνάρτησης

```
K>> dbstack
In Pat:Applications:V5:sqsum.m at line 4
In Pat:Applications:V5:variance.m at line 3
```

Ελέγξτε την τιμή του i και tot . Μετά τον έλεγχο της tot δώστε `dbstep` για έλεγχο της επόμενης τιμής του i :

```
K>> I
i =
    1

K>> dbstep
5 end
K>> tot

tot =
    4
K>> dbstep
End of M-file function Pat:Applications:V5:sqsum.m
```

Η συνάρτηση διέρχεται μέσα από το βρόχο μόνο μία φορά. Κοιτάζοντας την δήλωση :

```
for i = 1:length(mu)
```

είναι φανερό ότι υπάρχει λάθος σε αυτή την γραμμή.

Για να δούμε αν η αλλαγή της tot παράγει πλέον το σωστό αποτέλεσμα δώστε :

```
K>> tot = 10

tot =
10
```

Χρησιμοποιείστε την `dbup` για μετακίνηση στο χώρο εργασίας της `variance`. Είναι φανερό ότι η συνάρτηση δεχόταν την επιστρεφόμενη τιμή 4 για την μεταβλητή tot , την διαιρούσε με `length(x)-1`, το οποίο ήταν 4 και έδινε το λάθος αποτέλεσμα 1. Δώστε `dbcont` για έλεγχο πλέον της σωστής απόδοσης:

```
K>> dbup
In workspace belonging to Pat:Applications:V5:variance.m.

K>> dbcont

ans =
2.5000
```

9.4.6 Τερματισμός διαδικασίας διόρθωσης προγράμματος

Χρησιμοποιήστε την εντολή `dbquit` για τερματισμό της διαδικασίας διόρθωσης προγράμματος. Διορθώστε το `sqsum.m` ώστε ο επαναληπτικός βρόχος να ξεκινά από 1 μέχρι `length(mu)` :

```
for i = 1:length(x)
```

Στην συνέχεια με επανεκτέλεση της συνάρτησης παίρνουμε τα εξής αποτελέσματα :

```
variance(v)
```

```
ans =  
    2.5000
```

```
variance(w)
```

```
ans =  
   468.3876
```


10 ΑΝΑΓΝΩΣΗ - ΕΓΓΡΑΦΗ ΔΕΔΟΜΕΝΩΝ

Υπάρχουν αρκετοί τρόποι για να μεταφέρετε δεδομένα μεταξύ της MATLAB και άλλων εφαρμογών. Στις περισσότερες περιπτώσεις, μπορείτε να χρησιμοποιήσετε τις δυνατότητες της MATLAB για διάβασμα και δημιουργία αρχείων δεδομένων..

10.1 Εισαγωγή δεδομένων στην matlab

Μπορείτε να εισάγετε δεδομένα από αρχεία που έχουν κατασκευαστεί από διαφορετικά προγράμματα. Η MATLAB διαθέτει έναν μεγάλο αριθμό μεθόδων ανάγνωσης δεδομένων από αρχεία. Η επιλογή της καλύτερης μεθόδου εξαρτάται κάθε φορά από το μέγεθος και το είδος των δεδομένων.

1. Εισαγωγή δεδομένων ως κατάλογος στοιχείων. Αν έχουμε μικρό αριθμό δεδομένων, τότε είναι εύκολο να δώσουμε αυτές τις τιμές σαν δεδομένα πίνακα τοποθετώντας τα μέσα σε αγκύλες που συμβολίζουν τα όρια του πίνακα ([]). Αυτή η μέθοδος δεν χρησιμοποιείται για μεγάλο αριθμό δεδομένων επειδή αν κάνουμε κάποιο λάθος δεν μπορούμε να το διορθώσουμε απευθείας, αλλά θα πρέπει να χρησιμοποιήσουμε δηλώσεις θέσης του πίνακα στον οποίο τοποθετήσαμε τα δεδομένα.

2. Δημιουργία Δεδομένων σε M-file. Χρησιμοποιώντας έναν κειμενογράφο δημιουργούμε ένα M-file το οποίο περιέχει τις εντολές ανάθεσης τιμών σε πίνακα ή πίνακες. Η μέθοδος αυτή είναι χρήσιμη όταν τα δεδομένα δεν είναι διαθέσιμα σε ηλεκτρονική μορφή. Σε αντίθεση με την πρώτη μέθοδο, εδώ έχουμε το πλεονέκτημα της διόρθωσης των σφαλμάτων ανάθεσης τιμών στους πίνακες χρησιμοποιώντας τον κειμενογράφο. Με το πέρας των διορθώσεων εκτελούμε από το περιβάλλον εργασίας το M-file οπότε επιτυγχάνουμε την εκ νέου εισαγωγή των δεδομένων.

3. Φορτώνοντας δεδομένα από ένα ASCII αρχείο. Ένα τέτοιου είδους αρχείο περιέχει δεδομένα σε μορφή ASCII, με την κάθε γραμμή να έχει τον ίδιο αριθμό τιμών, με διαχωρισμό των τιμών με κενά καθώς επίσης και με την ύπαρξη του χαρακτήρα νέας γραμμής στο τέλος της. Η ανάγνωση αυτών των δεδομένων μπορεί να γίνει με την εντολή και συνάρτηση `load`, είτε τις εντολές χαμηλού επιπέδου `fscanf`, `fgets`, `gfetl`.

4. Ανάγνωση δυαδικών δεδομένων. Η ανάγνωση δυαδικών δεδομένων μπορεί να γίνει με την βοήθεια της εντολής `load` και την συνάρτηση `fread`. Αυτή η μέθοδος είναι χρήσιμη για το διάβασμα δεδομένων από άλλες εφαρμογές που χρησιμοποιούν δικές τους μορφοποιήσεις. Οι εντολές και συναρτήσεις οι οποίες διαβάζουν δυαδικά δεδομένα εκτελούνται ταχύτερα από τις αντίστοιχες εντολές και συναρτήσεις που διαβάζουν ASCII δεδομένα.

5. Χρησιμοποιώντας ειδικές συναρτήσεις διαβάσματος. Ο κατάλογος που ακολουθεί περιγράφει συναρτήσεις οι οποίες χρησιμοποιούνται για να διαβάζουν δεδομένα με ειδική μορφοποίηση:

```

dlmread  Ανάγνωση αρχείου ASCII δεδομένων.
wklread  Ανάγνωση αρχείου spreadsheet (WK1).
imread   Ανάγνωση αρχείου ψηφιακής εικόνας.
auread   Ανάγνωση αρχείου ήχου SUN (".au").
wavread  Ανάγνωση αρχείου ήχου της Microsoft WAVE
          (".wav").

```

6. Με την δημιουργία αρχείων MEX. Αυτή η μέθοδος είναι η πλέον αποδοτική σε ταχύτητα ανάκλησης δεδομένων αν υπάρχουν ήδη οι ρουτίνες σε C ή FORTRAN.

7. Δημιουργία προγράμματος σε C ή FORTRAN. Στην περίπτωση αυτή χρησιμοποιούμε προγράμματα που είναι γραμμένα σε C ή FORTRAN για τη μεταφορά των δεδομένων σε αρχεία τύπου MAT και στην συνέχεια χρησιμοποιούμε την διαταγή load της MATLAB.

10.2 Αποθήκευση δεδομένων από το περιβάλλον εργασίας της matlab σε αρχεία

Υπάρχουν αρκετοί μέθοδοι για εξαγωγή δεδομένων από την MATLAB και την αποθήκευση τους σε αρχεία. Οι σημαντικότερες από αυτές είναι οι ακόλουθες:

1. Για μικρούς πίνακες χρησιμοποιούμε την εντολή *diary*.
2. Αποθήκευση δεδομένων σε ASCII μορφή. Χρησιμοποιούμε την επιλογή *-ascii*
3. Αποθήκευση δεδομένων σε ειδική μορφή, χρησιμοποιώντας τις *fprintf*, *fwrite*, καθώς επίσης και άλλες συναρτήσεις χαμηλού επιπέδου I/O. Αυτή η μέθοδος είναι χρήσιμη για εγγραφή δεδομένων με την μορφή που απαιτούν άλλες εφαρμογές.
4. Χρησιμοποίηση ειδικών συναρτήσεων, όπως οι *wklwrite*, *dlmwrite*, *wavwrite* ή *imwrite*.

```

dlmwrite  Αποθήκευση σε αρχείο ASCII
wklwrite  Αποθήκευση σε αρχείο spreadsheet (WK1)
imwrite   Αποθήκευση πίνακα εικόνας σε αρχείο γραφικών
auwrite   Αποθήκευση πίνακα ήχου σε αρχείο SUN (".au")
wavwrite  Αποθήκευση πίνακα ήχου σε αρχείο Microsoft WAVE
          (".wav")

```

5. Ανάπτυξη αρχείων MEX. Αυτή είναι η καλύτερη μέθοδος εάν ήδη υπάρχουν οι υπορουτίνες με τις οποίες γίνονται οι εγγραφές των δεδομένων στην μορφή με την οποία απαιτούν άλλες εφαρμογές.

6. Εγγραφή των δεδομένων σε αρχεία MAT. Χρησιμοποιούμε την εντολή *save* για την δημιουργία ενός αρχείου MAT και στην συνέχεια δημιουργείται ένα πρόγραμμα σε C ή FORTRAN που μετατρέπει το αρχείο MAT στο κατάλληλο format.

10.3 Χαρακτήρες διαχωρισμού δεδομένων

Οι συναρτήσεις `dlmwrite`, `dlmread`, μας επιτρέπουν να αποθηκεύουμε σε αρχεία και να διαβάζουμε δεδομένα πίνακα οι τιμές των οποίων διαχωρίζονται με ειδικούς χαρακτήρες (delimiters separators).

Για παράδειγμα ας θεωρήσουμε ένα αρχείο το οποίο ονομάζεται "ph.dat" και στο οποίο τα δεδομένα χωρίζονται με (;):

```
7.2;8.5;6.2;6.6
5.4;9.2;8.1;7.2
```

Για να διαβάσουμε τα δεδομένα και να τα αποθηκεύσουμε σε έναν πίνακα A δίνουμε την ακόλουθη εντολή:

```
A = dlmread('ph.dat', ';');
```

Το δεύτερο όρισμα της συνάρτησης προσδιορίζει τον διαχωριστή. Ομοίως η συνάρτηση `dlmwrite`, αποθηκεύει δεδομένα σε αρχεία με τοποθέτηση ειδικών διαχωριστικών συμβόλων μεταξύ των δεδομένων

Αν ο πίνακας A περιέχει τα ακόλουθα δεδομένα

```
A =
      1      2      3
      4      5      6
```

και εκτελέσουμε την συνάρτηση

```
dlmwrite('myfile',A, ';')
```

το αρχείο myfile που δημιουργείτε περιέχει τα ακόλουθα ASCII δεδομένα:

```
1;2;3
4;5;6
```

11 ΣΥΝΑΡΤΗΣΕΙΣ ΠΟΥ ΔΕΧΟΝΤΑΙ ΟΝΟΜΑΤΑ ΣΥΝΑΡΤΗΣΕΩΝ

Σε αυτό το κεφάλαιο περιγράφουμε συναρτήσεις της MATLAB που δέχονται σε ένα τουλάχιστον από τα ορίσματά τους όνομα μαθηματικών συναρτήσεων είτε όνομα συνάρτησης που έχει οριστεί σε ένα M-file. Η ομάδα αυτών των συναρτήσεων που θα την ονομάζουμε στο εξής συναρτήσεις συναρτήσεων περιέχουν δυνατότητες:

- Αριθμητικής ολοκλήρωσης
- Βελτιστοποίησης και λύσης μη γραμμικών εξισώσεων
- Επίλυση διαφορικών εξισώσεων
- Γραφικών απεικονίσεων

Οι συναρτήσεις συνάρτησης βρίσκονται στον κατάλογο `funfun` ο οποίος περιέχεται σαν Toolbox και διανέμεται από την MathWorks, την εταιρεία που κατασκευάζει την MATLAB.

Όνομα συνάρτησης	Συνοπτική περιγραφή εντολών
<code>fmin</code>	Ελαχιστοποίηση συνάρτησης μιας μεταβλητής
<code>fmins</code>	Ελαχιστοποίηση συνάρτησης πολλών μεταβλητών
<code>fzero</code>	Εύρεση ρίζας μιας μεταβλητής
<code>quad</code>	Αριθμητική ολοκλήρωση συνάρτησης με μέθοδο χαμηλής τάξης
<code>quad8</code>	Αριθμητική ολοκλήρωση συνάρτησης με μέθοδο υψηλής τάξης
<code>dblquad</code>	Αριθμητική ολοκλήρωση συνάρτησης δύο μεταβλητών
<code>ezplot</code>	Γραφική παράσταση συνάρτησης μιας μεταβλητής (ή εύκολη έκδοση)
<code>fplot</code>	Γραφική παράσταση συνάρτησης μιας μεταβλητής, (η απλή έκδοση)

Η MATLAB έχει την δυνατότητα να αναπαριστά μαθηματικές συναρτήσεις δίνοντας την μέθοδο υπολογισμού της σε M-files.

Έστω ότι έχουμε την εξής συνάρτηση:

$$f(x) = \frac{1}{(x-0.3)^2 + 0.01} + \frac{1}{(x-0.9)^2 + 0.04} - 6$$

Αυτή η συνάρτηση μπορεί να χρησιμοποιηθεί σαν είσοδος σε οποιαδήποτε άλλη από τις συναρτήσεις συνάρτησης. Είναι ένα M-file με την ονομασία `humps.m`:

```
function y = humps(x)
y = 1./((x - 0.3).^2 + 0.01) + 1./((x - 0.9).^2 + 0.04) - 6;
```

Όλες οι συναρτήσεις που περιγράφονται σε αυτό το κεφάλαιο ονομάζονται συναρτήσεις συνάρτησης γιατί δέχονται σαν ένα από τα ορίσματά τους το όνομα ενός M-file όπως αυτό του `humps` που ορίζει μια μαθηματική συνάρτηση.

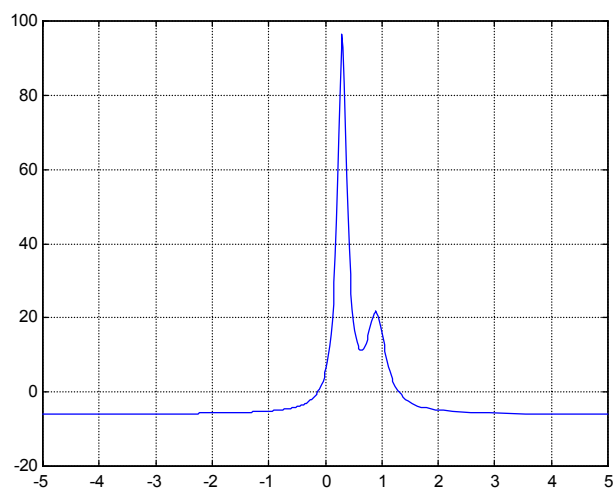
11.1 Γραφική παράσταση μαθηματικών συναρτήσεων

fplot

Η συνάρτηση `fplot` δίνει την γραφική παράσταση μιας μαθηματικής συνάρτησης μεταξύ δοσμένων ορίων του άξονα. Μπορεί κανείς να θέσει τα όρια στον άξονα x ή και στους δύο άξονες.

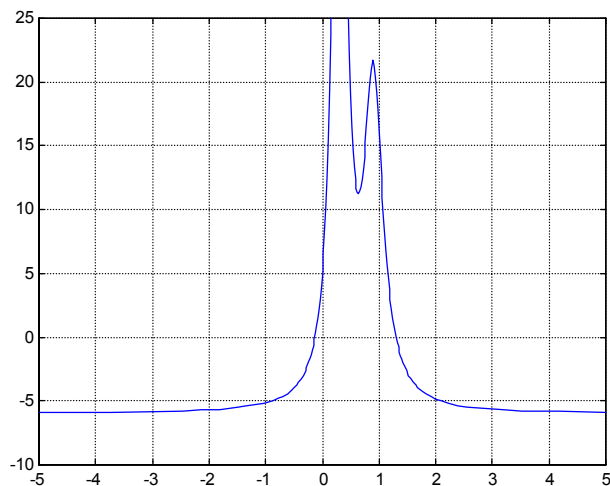
Για να λάβουμε την γραφική παράσταση της μαθηματικής συνάρτησης `humps` στον άξονα x στο διάστημα $[-5\ 5]$ χρησιμοποιούμε τις εξής εντολές:

```
fplot('humps', [-5 5])
grid on
```



Μπορούμε να λάβουμε λεπτομερέστερη ανάλυση της γραφικής παράστασης της συνάρτησης διαλέγοντας τα όριά της στον άξονα y από -10 μέχρι 25 , χρησιμοποιώντας τις ακόλουθες εντολές

```
fplot('humps', [-5 5 -10 25])
grid on
```



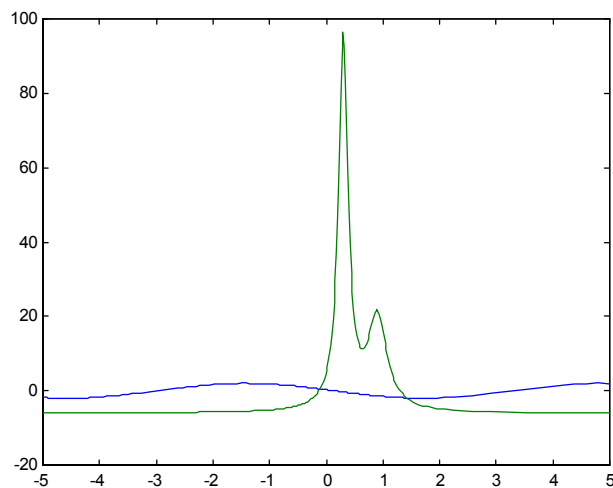
Επίσης η έκφραση μιας μαθηματικής συνάρτησης μπορεί να γραφεί απευθείας στην `fplot` όπως φαίνεται παρακάτω :

```
fplot('2*sin(x+3)', [-1 1])
```

Η MATLAB δίνει την δυνατότητα γραφικής αναπαράστασης περισσότερων από δύο συναρτήσεων στην ίδια γραφική παράσταση με χρήση της `fplot` μόνο μια φορά. Αν χρησιμοποιήσουμε την `fplot` για περισσότερες από μια συναρτήσεις τότε αυτή πρέπει να γραφεί με μορφή πίνακα του οποίου κάθε στήλη περιέχει τις τιμές της συνάρτησης στο πεδίο ορισμού που τίθεται στην συνάρτηση.

Στο επόμενο παράδειγμα έχουμε ταυτόχρονη γραφική παράσταση δύο μαθηματικών συναρτήσεων ($2*\sin(x+3)$, `humps`). Η σύνταξη της αντίστοιχης εντολής είναι:

```
fplot('[2*sin(x+3), humps(x)]', [-5 5])
```



Όπως παρατηρούμε η έκφραση `[2*sin(x+3), humps(x)]` υπολογίζει κάθε μια συνάρτηση σε ένα πίνακα δύο στηλών όταν το x είναι άνυσμα στήλης.

12 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΕ M-FILES

12.1 Δημιουργία M-Files

Τα M-Files είναι κοινά αρχεία κειμένου (ASCII) τα οποία μπορούν να δημιουργηθούν με την βοήθεια επεξεργαστών κειμένου. Η MATLAB έχει ενσωματωμένο δικό της επεξεργαστή κειμένου ο οποίος μπορεί να ξεκινήσει από το *File* μενού με επιλογή του *New* και μετά *M-File*.

edit

Ένας άλλος τρόπος για την δημιουργία ενός αρχείου είναι χρησιμοποιώντας την εντολή `edit`:

```
edit <όνομα αρχείου>
```

Η εντολή αυτή ανοίγει τον επεξεργαστή κειμένου της MATLAB για το αρχείο <όνομα αρχείου>. Χρησιμοποιώντας την εντολή `edit` χωρίς όνομα αρχείου:

```
edit
```

το περιβάλλον εργασίας ανοίγει τον επεξεργαστή κειμένου σε `untitled file`.

12.2 Scripts

Τα scripts είναι η πιο απλή μορφή των M-Files, δεν περιέχουν μεταβλητές εισόδου, ούτε δίνουν μεταβλητές εξόδου αλλά μπορούν να χρησιμοποιήσουν και να μεταβάλουν τις μεταβλητές που έχουν οριστεί στον χώρο εργασίας. Είναι χρήσιμα για την αυτοματοποίηση πολλαπλών εντολών που δίνονται από το πληκτρολόγιο. Όλες οι μεταβλητές που δημιουργούνται από τα scripts παραμένουν στο χώρο εργασίας και μετά το τέλος της εκτέλεσης τους και μπορούν να χρησιμοποιηθούν από τον χρήστη.

12.3 Συναρτήσεις

Οι συναρτήσεις είναι M-Files τα οποία δέχονται μεταβλητές εισόδου και αποδίδουν μεταβλητές εξόδου. Οι μεταβλητές που ορίζονται μέσα στις συναρτήσεις είναι τοπικές μεταβλητές. Κατά συνέπεια όταν ο έλεγχος του προγράμματος φύγει από την συνάρτηση, όλες οι τοπικές μεταβλητές διαγράφονται. Οι μεταβλητές των συναρτήσεων έχουν το δικό τους χώρο εργασίας.

```
Function f = fact(n)
% FACT Factorial.
% FACT(N) returns the factorial of N, usually denoted by N!

f = prod(1:n) ;
```

Αν η συνάρτηση έχει περισσότερα από ένα ορίσματα εξόδου, τότε θα πρέπει να

κλείνονται σε αγκύλες (`[]`). Τα δεδομένα εισόδου, αν υπάρχουν, πρέπει να περικλείονται σε παρενθέσεις. Για τον διαχωρισμό των δεδομένων είτε της εισόδου είτε της εξόδου χρησιμοποιούμε το κόμμα (,). Δείτε το παράδειγμα που ακολουθεί:

```
function [x, y, z] = sphere ( theta, phi, rho)
```

Αν δεν υπάρχουν δεδομένα εξόδου, τότε χρησιμοποιούμε κενές αγκύλες

```
function [] = printresults ( x )
```

ή τίποτα:

```
function printresults ( x )
```

Οι μεταβλητές που περνιούνται ως ορίσματα σε μια συνάρτηση δεν είναι απαραίτητο να έχουν το ίδιο όνομα με αυτές που αναφέρονται στην γραμμή ορισμού.

Κάθε συνάρτηση έχει κάποια κοινά στοιχεία που συναντώνται σε όλες τις συναρτήσεις της MATLAB:

- **Η γραμμή δήλωσης της συνάρτησης.** Σε αυτή την γραμμή δηλώνεται το όνομα της συνάρτησης και ο αριθμός και η σειρά των ορισμάτων εισόδου και εξόδου.
- **Η γραμμή help.** Αυτή η γραμμή εμφανίζεται κάθε φορά που χρησιμοποιείται η εντολή `lookfor` ή `help` για αυτή την συνάρτηση.
- **Το κείμενο της help.** Εμφανίζεται μαζί με την γραμμή `help` όποτε εκτελείται η εντολή `help` για την συνάρτηση.
- **Ο κώδικας της συνάρτησης.** Είναι ο κώδικας που εκτελεί το υπολογιστικό τμήμα της συνάρτησης.

Τα ονόματα αρχείων στα οποία μπορούν να αποθηκευτούν οι συναρτήσεις στην MATLAB υπόκεινται στους ίδιους περιορισμούς με αυτούς των μεταβλητών. Η MATLAB χρησιμοποιεί τους πρώτους 31 χαρακτήρες των ονομάτων. Όλα τα ονόματα των συναρτήσεων πρέπει να ξεκινούν με γράμμα, τα υπόλοιπα μπορούν να είναι συνδυασμός είτε αριθμών, γραμμάτων κλπ. Αν το όνομα του αρχείου με το οποίο έχει αποθηκευτεί μια συνάρτηση είναι διαφορετικό από το συμβολικό όνομα της συνάρτησης, τότε το όνομα αποθήκευσης είναι ισχυρότερο και το εσωτερικό όνομα της συνάρτησης αγνοείται. Για να αποφεύγουμε λάθη κατανόησης των προγραμμάτων προτείνεται η χρήση ιδίου ονόματος για το συμβολικό όνομα της συνάρτησης και του αρχείου στο οποίο αποθηκεύεται.

nargin, nargout

Οι συναρτήσεις `nargin`, `nargout` χρησιμοποιούνται για να υπολογιστεί ο αριθμός των ορισμάτων εισόδου και εξόδου αντίστοιχα που έχουν δοθεί σε μια συνάρτηση. Με την βοήθεια λογικών εντολών και των συναρτήσεων `nargin`, `nargout` μπορούμε να κατασκευάσουμε κώδικα ο οποίος να εκτελεί διαφορετικές εντολές ανάλογα με τον αριθμό των ορισμάτων που έχουν δοθεί:

```
function c = testarg1(a,b)
if (nargin == 1)
```



```

    c = a.^2 ;
elseif (nargin == 2)
    c = a+b ;
end

```

Από το πιο πάνω παράδειγμα φαίνεται ότι αν δοθεί μόνο ένα όρισμα εισόδου, τότε το αποτέλεσμα της συνάρτησης θα είναι ο υπολογισμός του τετραγώνου του ορίσματος, ενώ αν δοθούν δύο ορίσματα εισόδου, το αποτέλεσμα θα είναι να υπολογιστεί το άθροισμα τους.

varargin, varargout

Οι συναρτήσεις `varargin` και `varargout` επιτρέπουν την διαχείριση μεταβλητού αριθμού ορισμάτων εισόδου ή την επιστροφή οποιουδήποτε αριθμού ορισμάτων εξόδου αντίστοιχα από μια συνάρτηση. Η MATLAB δεσμεύει όλα τα δεδομένα εισόδου και εξόδου σε ειδικούς πίνακες με το όνομα *cell arrays* οι οποίοι περιέχουν στοιχεία για άλλους πίνακες. Κάθε ένα στοιχείο αυτών των πινάκων μπορεί να δεσμεύσει οποιοδήποτε μέγεθος διανύσματος ή αριθμητικών δεδομένων. Για παράδειγμα η παρακάτω συνάρτηση δέχεται ως είσοδο οποιονδήποτε αριθμό από σημεία στον χώρο και κατασκευάζει μια γραφική παράσταση για αυτά:

```

function testvar(varargin)
for i = 1 : length(varargin)
    x(i) = varargin{i}(1) ;
    y(i) = varargin{i}(2) ;
end
xmin = min(0, min(x)) ;
ymin = min(0, min(y)) ;
axis([xmin fix(max(x))+3 ymin fix(max(y))+3])
plot(x,y)

```

Ο ορισμός της συνάρτησης `testvar` που δόθηκε μπορεί να χρησιμοποιήσει μεταβλητό αριθμό δεδομένων εισόδου:

```

testvar ([2 3], [1 5], [4 8], [6 5], [4 2], [2 3])
testvar ([-1 0], [3 -5], [4 2], [1 1])

```

Επειδή η μεταβλητή `varargin` αποθηκεύει όλα τα δεδομένα εισόδου σε έναν πίνακα *cell array* είναι απαραίτητο να χρησιμοποιηθεί διαφορετική δεικτοδότηση για την εξαγωγή των δεδομένων. Για παράδειγμα:

```

y(i) = varargin{i}(2) ;

```

Στο πιο πάνω παράδειγμα φαίνονται δύο δείκτες:

- Ο δείκτης κελιού μέσα σε αγκύλες και
- Ο δείκτης στοιχείου μέσα στο κελί σε παρενθέσεις.

Όταν χρησιμοποιείται το `varargout` θα πρέπει, όπως και με την `varargin`, όλα τα δεδομένα εξόδου να τοποθετηθούν σε ένα πίνακα κελιών. Με την χρήση της εντολής `nargout` υπολογίζεται ο αριθμός των δεδομένων εξόδου αυτής της συνάρτησης. Για παράδειγμα ο πιο κάτω κώδικας δέχεται σαν είσοδο ένα πίνακα με δύο στήλες. Η πρώτη στήλη αντιπροσωπεύει την διάσταση x και η δεύτερη στήλη την διάσταση y . Κατόπιν

σπάει τον πίνακα σε ξεχωριστά διανύσματα $[x_i \ y_i]$ τα οποία μπορούν να δοθούν ως δεδομένα εισόδου στην προηγούμενη συνάρτηση `testvar`:

```
function [varargout] = testvar2 (arrayin)
for i = 1:nargout
    varargout{i} = arrayin(i,:);
end
```

Η κλήση της πιο πάνω συνάρτησης γίνεται ως εξής:

```
a = [1 2 3 4 5 ; 6 7 8 9 0]' ;
[p1, p2, p3, p4, p5] = testvar2(a)
```

Προσοχή πρέπει να δοθεί στο σημείο αυτό ότι τα `varargin`, `varargout` πρέπει πάντοτε να δίνονται τελευταία σαν ορίσματα των συναρτήσεων δηλαδή:

```
function [out1, out2] = example1(a, b, varargin)
function [i, j, varargout] = example2 (x1, y1, x2, y2, flag)
```

12.4 Υποσυναρτήσεις

Οι συναρτήσεις που είναι γραμμένες σε M-Files εκτός από την *κύρια συνάρτηση* μπορεί να περιέχουν και άλλες *υποσυναρτήσεις* γραμμένες στο ίδιο αρχείο. Αυτές οι *υποσυναρτήσεις* είναι ορατές μόνο μέσα από την *κύρια συνάρτηση* και δεν μπορούν να κληθούν έξω από αυτήν. Κάθε *υποσυνάρτηση* ξεκινάει με την δικιά της γραμμή ορισμού συνάρτησης. Οι *υποσυναρτήσεις* μπορούν να εμφανίζονται με οποιαδήποτε σειρά μέσα σε ένα αρχείο αρκεί η κύρια συνάρτηση να εμφανίζεται πάντα πρώτη. Οι συναρτήσεις που βρίσκονται μέσα στο ίδιο M-File δεν μπορούν να προσπελάσουν τις ίδιες μεταβλητές εκτός και αν αυτές δηλωθούν ως καθολικές συναρτήσεις ή περαστούν σαν ορίσματα συναρτήσεων

```
function [avg,med] = newstats(u)
% NEWSTATS Find mean and median with internal functions.
n = length(u);
avg = mean(u,n);
med = median(u,n);

function a = mean(v,n)
% Calculate average.
a = sum(v)/n;

function m = median(v,n)
% Calculate median.
w = sort(v);
if rem(n,2) == 1
    m = w((n+1)/2);
else
    m = (w(n/2)+w(n/2+1))/2;
end
```

Από το παραπάνω παράδειγμα φαίνεται ότι η κύρια συνάρτηση είναι η `newstats` η οποία υπολογίζει την μέση τιμή και τον μέσο στοιχείο μιας σειράς αριθμών. Ο υπολογισμός της μέσης τιμής και του μέσου γίνεται από δύο υποσυναρτήσεις.

12.5 Λογικές συναρτήσεις

Εκτός από τους λογικούς τελεστές, η MATLAB διαθέτει και μερικές λογικές συναρτήσεις. Οι σημαντικότερες αυτών είναι οι `xor`, `all`, `any`, `isnan` και `isinf`. Οι συναρτήσεις αυτές χρησιμοποιούνται για να ελέγχουν το περιεχόμενο πινάκων όπως και το αποτέλεσμα αριθμητικών πράξεων το οποίο έχει αποθηκευτεί σε μεταβλητές.

`xor, all, any, isnan, isinf`

- Η συνάρτηση `xor`, εκτελεί το λογικό `xor` σε μεταβλητές, διανύσματα ή πίνακες.
- Η συνάρτηση `all`. Αυτή η συνάρτηση επιστρέφει την αριθμητική τιμή 1 αν όλα τα στοιχεία ενός διανύσματος έχουν μη-μηδενικό περιεχόμενο.
- Η συνάρτηση `any`. Επιστρέφει την αριθμητική τιμή 1 αν υπάρχει κάποιο στοιχείο του διανύσματος με μη μηδενική τιμή, διαφορετικά επιστρέφει 0. Η συνάρτηση `any` αυτή εφαρμόζεται σε κάθε στήλη ενός πίνακα όπως και η `all`.
- Η συνάρτηση `isnan` και `isinf`. Επιστρέφει την αριθμητική τιμή 1 όταν το περιεχόμενο μεταβλητής έχει λάβει την τιμή των σταθερών `NaN` και `Inf`.

`find`

Η συνάρτηση `find` επιστρέφει τις θέσεις του πίνακα που ικανοποιούν την λογική έκφραση που τοποθετείται.

Με τις εντολές που ακολουθούν δίνουμε ακέραιες τιμές στον πίνακα `A`.

```
A = magic(4)
```

```
A =
```

```
16  2  3 13
 5 11 10  8
 9  7  6 12
 4 14 15  1
```

Με την εντολή που ακολουθεί δημιουργούμε τον πίνακα `i` που περιέχει τις θέσεις του πίνακα `A` για τις οποίες το περιεχόμενό τους είναι μεγαλύτερο του 8. Στην συνέχεια τοποθετείται η τιμή 100 σε αυτές τις θέσεις του πίνακα.

```
i = find(A > 8);
A(i) = 100
```

```
A =
```

```

100    2    3   100
   5 100  100    8
100    7    6   100
   4 100  100    1

```

12.6 Εκτέλεση εντολών που περιέχονται σε μεταβλητή χαρακτήρων

Μια πολύ ενδιαφέρουσα κατηγορία συναρτήσεων αναφέρεται σε συναρτήσεις οι οποίες έχουν την δυνατότητα να εκτελούν εντολές οι οποίες είναι τοποθετημένες μέσα σε μεταβλητές χαρακτήρων. Η δυνατότητα αυτή είναι πολύ χρήσιμη όταν θέλουμε να εκτελούμε εντολές οι οποίες σχηματίζονται κατά την διάρκεια εκτέλεσης του προγράμματος.

eval

Η συνάρτηση `eval` εκτελεί την εντολή που υπάρχει στην μεταβλητή κειμένου. Η έκφραση μπορεί να περιέχει μια έκφραση της MATLAB, μια δήλωση ή μια κλήση συνάρτησης. Η πιο απλή μορφή κλήσης της συνάρτησης είναι:

```
eval ('<αλφαριθμητικό>')
```

Για παράδειγμα ο παρακάτω κώδικας χρησιμοποιεί την `eval` για να δημιουργήσει ένα πίνακα Hilbert τάξης `n`:

```

t = '1/(i+j-1)';
for i = 1:n
for j = 1:n
a(i,j) = eval(t);
end
end

```

feval

Η συνάρτηση `feval` διαφέρει από την προηγούμενη συνάρτηση `eval` διότι χρησιμοποιείται για την εκτέλεση συναρτήσεων και όχι εκφράσεων της MATLAB. Αποτελεί δηλαδή ειδική περίπτωση της γενικότερης συνάρτησης `eval`.

Όπως και με τους πίνακες, έτσι και με τις μεταβλητές κειμένου, μπορούμε να ενώσουμε πολλές μεταβλητές για την δημιουργία ενός μεγαλύτερου κειμένου:

```

for i=1:10
    eval(['P',int2str(i),'= i.^2'])
end

```

Το πιο πάνω πρόγραμμα έχει ως αποτέλεσμα την δημιουργία των μεταβλητών κειμένου `P1`, `P2`, ..., `P10` στις οποίες τίθενται διαφορετικές τιμές.

12.7 Άδειοι πίνακες

Για την δημιουργία άδειων πινάκων μπορεί να χρησιμοποιηθεί το διπλό άγκιστρο (`[]`). Επίσης άδειοι πίνακες μπορούν να δημιουργηθούν (αρχικοποιηθούν) χρησιμοποιώντας και την εντολή της MATLAB `zeros`:

```
E = zeros(0,5)
```

12.8 Εισαγωγή δεδομένων από τον χρήστη

Για την εισαγωγή δεδομένων από τον χρήστη, η MATLAB χρησιμοποιεί δύο τρόπους:

- Προβολή μηνύματος και αναμονή για είσοδο των δεδομένων από το πληκτρολόγιο
- Παύση του προγράμματος μέχρι ο χρήστης να πιάσει κάποιο πλήκτρο.

input

Η σύνταξη της εντολής `input` είναι:

```
n = input ('<αλφαριθμητικό>')
```

Όπου `αλφαριθμητικό` είναι το κείμενο το οποίο εμφανίζεται κατά την αναμονή πληκτρολόγησης από τον χρήστη του τερματικού και `n` είναι η μεταβλητή που περιέχει τα δεδομένα που έδωσε ο χρήστης από το πληκτρολόγιο του τερματικού. Υπάρχει δυνατότητα να λάβουμε και το `αλφαριθμητικό` κείμενο που εισήγαγε ο χρήστης με την εντολή που ακολουθεί:

```
name = input ('Εισάγετε το όνομα σας: ', 's') ;
```

Σε αυτή την περίπτωση το `αλφαριθμητικό` κείμενο που εισήγαγε ο χρήστης αποθηκεύεται στην μεταβλητή `s`.

pause

Η εντολή `pause` σταματάει προσωρινά την εκτέλεση του προγράμματος μέχρι ο χρήστης να πιάσει κάποιο πλήκτρο. Ειδικότερα η εντολή `pause(n)` σταματά την εκτέλεση του προγράμματος στο σημείο που βρίσκεται τοποθετημένη η εντολή για `n` δευτερόλεπτα.

12.9 Βελτιστοποίηση της ταχύτητας υπολογισμών

Επειδή η MATLAB είναι μια γλώσσα προσανατολισμένη σε πράξεις με πίνακες, υπάρχει τρόπος να βελτιστοποιηθεί η απόδοση του κώδικα αν αντί για βρόχους `for` χρησιμοποιηθούν αντίστοιχα πράξεις πινάκων.

Ένας δεύτερος τρόπος βελτίωσης του χρόνου εκτέλεσης των προγραμμάτων είναι με προ-δέσμευση των διαστάσεων των πινάκων που θα χρησιμοποιηθούν από το πρόγραμμα. Αυτό έχει το σημαντικό πλεονέκτημα να μην χρειάζεται η MATLAB να

κάνει επαναπροσδιορίζει τις διαστάσεις των πινάκων κάθε φορά που μεταβάλλεται το μέγεθος τους. Την τεχνική αυτή θα την αναφέρουμε στο τμήμα των σημειώσεων που ακολουθεί.

Για παράδειγμα έστω ότι θέλουμε να υπολογίσουμε το ημίτονο των 1001 τιμών από το 0 μέχρι το 10. Ο πρώτος τρόπος με την βοήθεια των βρόχων `for` είναι:

```
i = 0;
for t = 0:.01:10
    i = i+1;
    y(i) = sin(t);
end
```

Το ίδιο αποτέλεσμα σε πιο απλή μορφή μπορεί να επιτευχθεί χρησιμοποιώντας πίνακες:

```
t = 0:.01:10;
y = sin(t);
```

Το αποτέλεσμα του κώδικα της δεύτερης υλοποίησης είναι η εκτέλεση του προγράμματος να είναι γρηγορότερη σε σχέση με την πρώτη υλοποίηση.

`tic, toc`

Για να μετρήσετε τον χρόνο που χρειάζεται να τρέξουν τα προγράμματα μπορούν να χρησιμοποιηθούν οι εντολές `tic` και `toc`.

12.10 Διαχείριση μνήμης

Η MATLAB έχει πέντε συναρτήσεις για την διαχείριση της μνήμης του υπολογιστή:

- `clear` διαγράφει μεταβλητές από την μνήμη
- `pack` αποθηκεύει το περιεχόμενο όλων των μεταβλητών στον δίσκο, διαγράφει όλες τις μεταβλητές του προγράμματος και στην συνέχεια διαβάζει ξανά τα δεδομένα από τον δίσκο του υπολογιστή. Η διαδικασία αυτή διαγράφει άχρηστες μεταβλητές και περιοχές της μνήμης οι οποίες δεν χρησιμοποιούνται.
- `quit` σταματά την εκτέλεση του μεταγλωττιστή της MATLAB και επιστρέφει όλη τη δεσμευμένη μνήμη στο λειτουργικό σύστημα.
- `save` αποθηκεύει επιλεκτικά κάποιες μεταβλητές σε αρχείο.
- `load` διαβάζει τις μεταβλητές που έχουν αποθηκευτεί με την εντολή `save`.

13 ΠΙΝΑΚΕΣ ΧΑΡΑΚΤΗΡΩΝ

Αυτό το κεφάλαιο περιγράφει τους δύο τρόπους με τους οποίους γίνεται η διαχείριση δεδομένων κειμένου ή αλφαριθμητικών δεδομένων στην MATLAB. Περιγράφονται οι εντολές και συναρτήσεις δημιουργίας πινάκων χαρακτήρων και η δημιουργία πινάκων που περιέχουν σε κάθε θέση τους ένα χαρακτήρα. Επίσης αναφέρονται οι βασικές αλφαριθμητικές εντολές, όπως η αναζήτηση και αντικατάσταση κειμένων και πώς γίνεται η μετατροπή μεταξύ αλφαριθμητικών και αριθμητικών μορφοποιήσεων.

Στη MATLAB, ο όρος αλφαριθμητικό αναφέρεται σε έναν πίνακα χαρακτήρων. Η MATLAB αναπαριστά κάθε χαρακτήρα εσωτερικά με την αντίστοιχη ASCII τιμή του. Τα δεδομένα των χαρακτήρων εισάγονται στην MATLAB με την βοήθεια εισαγωγικών (''). Για παράδειγμα:

```
name = 'Thomas R. Lee';

whos

ans =

Name      Size      Bytes      Class

name      1x13      26 char      array
```

Οι συναρτήσεις διαχείρισης κειμένων βρίσκονται στον κατάλογο `strfun` στο Toolbox της MATLAB.

Όνομα συνάρτησης	Συνοπτική περιγραφή εντολών
<code>char</code>	Δημιουργία πίνακα χαρακτήρων.
<code>double</code>	Μετατροπή αλφαριθμητικού σε αριθμητικούς κώδικες.
<code>cellstr</code>	Δημιουργία πίνακα κελιών αλφαριθμητικών από πίνακα χαρακτήρων.
<code>blanks</code>	κείμενα από κενά διαστήματα
<code>deblank</code>	Απομάκρυνση των κενών διαστημάτων που βρίσκονται στο τέλος του κειμένου
<code>eval</code>	Εκτέλεση αλφαριθμητικής έκφρασης της MATLAB .
<code>ischar</code>	Έλεγχος για ύπαρξη πίνακα χαρακτήρων
<code>iscellstr</code>	Έλεγχος για ύπαρξη πίνακα κειμένου.
<code>isletter</code>	Έλεγχος για ύπαρξη γραμμάτων του λατινικού αλφάβητου.
<code>isspace</code>	Έλεγχος για ύπαρξη κενών χαρακτήρων
<code>strcat</code>	Ένωση αλφαριθμητικών κειμένων
<code>strvcat</code>	Ένωση οριζόντιων αλφαριθμητικών κειμένων .
<code>strcmp</code>	Σύγκριση αλφαριθμητικών κειμένων
<code>strncmp</code>	Σύγκριση των πρώτων N χαρακτήρων κειμένων.
<code>findstr</code>	Εύρεση ενός αλφαριθμητικού κειμένου μέσα σε ένα άλλο.
<code>strjust</code>	Στοίχιση αλφαριθμητικού κειμένου
<code>strmatch</code>	Εύρεση ενός αλφαριθμητικού κειμένου μέσα σε ένα άλλο.

Όνομα συνάρτησης	Συνοπτική περιγραφή εντολών
<code>strrep</code>	Αντικατάσταση ενός κειμένου με κάποιο άλλο.
<code>strtok</code>	Εύρεση του λέξης κειμένου μέσα σε κείμενο.
<code>upper</code>	Μετατροπή κειμένου σε κεφαλαία λατινικά.
<code>lower</code>	Μετατροπή κειμένου σε μικρά λατινικά.
<code>num2str</code>	Μετατροπή αριθμού σε κείμενο.
<code>int2str</code>	Μετατροπή ακεραίου σε κείμενο.
<code>mat2str</code>	Εκτέλεση εντολής κειμένου και αποθήκευση του αποτελέσματος πάλι σε κείμενο
<code>str2num</code>	Μετατροπή κειμένου σε αριθμό.
<code>sprintf</code>	Μεταφορά και μετασχηματισμός δεδομένων σε κείμενο
<code>sscanf</code>	Μεταφορά και μετασχηματισμός κειμένου σε μεταβλητές
<code>hex2num</code>	Μετατροπή δεκαεξαδικού σε αριθμό διπλής ακρίβειας.
<code>hex2dec</code>	Μετατροπή δεκαεξαδικού σε δεκαδικό
<code>dec2hex</code>	Μετατροπή δεκαδικού σε δεκαεξαδικό αλφαριθμητικό.
<code>bin2dec</code>	Μετατροπή δυαδικού κειμένου σε δεκαδικό ακέραιο.
<code>dec2bin</code>	Μετατροπή δεκαδικού σε δυαδικό κείμενο
<code>base2dec</code>	Μετατροπή κειμένου οποιασδήποτε βάσης σε δεκαδικό ακέραιο
<code>dec2base</code>	Μετατροπή δεκαδικού ακέραιου σε κείμενο οποιασδήποτε βάσης

13.1 Συναρτήσεις κειμένου

Οι συναρτήσεις κειμένου διευκολύνουν σημαντικά τον χειρισμό μεταβλητών και πινάκων που περιέχουν αλφαριθμητικούς χαρακτήρες. Η χρήση τέτοιου είδους συναρτήσεων είναι περιορισμένη διότι MATLAB είναι προσανατολισμένη για αριθμητικούς υπολογισμούς πινάκων.

class, ischar

Η συνάρτηση `class` επιστρέφει τον τύπο της μεταβλητής, ενώ η συνάρτηση `ischar` επιστρέφει την τιμή 1 (αληθής) όταν η μεταβλητή του ορίσματος που δέχεται είναι πίνακας χαρακτήρων:

```
class(name)
```

```
ans =  
char
```

```
ischar(name)
```

```
ans =  
1
```


13.2 Μετατροπή από ASCII χαρακτήρες σε αριθμητικές τιμές του ASCII πίνακα

Οι πίνακες χαρακτήρων αποθηκεύουν κάθε χαρακτήρα σε πεδίο των 16bit. Κάθε χαρακτήρας διαθέτει μία μοναδική θετική ακέραια τιμή ASCII τιμή. Με την βοήθεια των συναρτήσεων που ακολουθούν μπορούμε να μετατρέψουμε τις αριθμητικές ASCII τιμές σε κείμενο και το αντίστροφο.

double, char

Με την βοήθεια της `double` οι χαρακτήρες του κειμένου μπορούν να μετατραπούν στις αντίστοιχες αριθμητικές τιμές τους, ενώ με την βοήθεια της `char` πετυχαίνουμε το αντίθετο αποτέλεσμα.

```
name = double (name)
```

```
name =
      84  93  34  2  2  4  55
```

```
name = char (name)
```

```
name =
      ksdjhfksdj
```

13.3 Δημιουργία δισδιάστατων πινάκων χαρακτήρων

Με τους δισδιάστατους πίνακες χαρακτήρων μπορούμε να αποθηκεύουμε πολλαπλά κείμενα. Όταν δημιουργούνται δισδιάστατοι πίνακες το μέγιστο μήκος κάθε κειμένου είναι ίδιο. Μικρότερα κείμενα τοποθετούνται στον πίνακα εισάγοντας κενούς χαρακτήρες μετά το τέλος του κειμένου έτσι ώστε να εξισώσουμε το μήκος όλων των αλφαριθμητικών μεταβλητών.

char

Ο πιο απλός τρόπος όμως για την δημιουργία πινάκων χαρακτήρων είναι χρησιμοποιώντας την εντολή `char`. Με αυτή την εντολή τα αλφαριθμητικά εξισώνουν αυτόματα το μήκος τους και προστίθενται αυτόματα οι απαραίτητοι κενοί χαρακτήρες στο τέλος του κάθε κειμένου. Έτσι το μήκος τους γίνεται ίσο με το μήκος του μεγαλύτερου αλφαριθμητικού.

```
name = char ('αλφαριθμητικό 1', 'αλφαριθμητικό2');
```

dblank

Όταν θέλουμε να πάρουμε κάποιο αλφαριθμητικό κείμενο από έναν πίνακα χαρακτήρων συνίσταται η χρήση της εντολής `dblank` η οποία διαγράφει όλα τα κενά από το τέλος της αλφαριθμητικής μεταβλητής.

cellstr

Το βασικό μειονέκτημα των πινάκων χαρακτήρων είναι το ότι χρειάζεται να έχουν όλα τα στοιχεία τους ίδιο μήκος. Η MATLAB έχει και ένα άλλο τύπο πινάκων, τους πίνακες κελιών χαρακτήρων, τα στοιχεία των οποίων δεν είναι απαραίτητο να έχουν το ίδιο μέγεθος. Η μετατροπή ενός πίνακα χαρακτήρων σε πίνακα κελιών χαρακτήρων γίνεται με την βοήθεια της εντολής `cellstr`. Η αντίθετη μετατροπή γίνεται με τη βοήθεια της συνάρτησης `char`.

13.4 Σύγκριση αλφαριθμητικών κειμένων

Η MATLAB έχει δύο συναρτήσεις οι οποίες μπορούν να χρησιμοποιηθούν για να αποφανθούν αν δύο αλφαριθμητικά είναι ίσα ή όχι, την `strcmp` και την `strncmp`.

strcmp, strncmp

Οι συναρτήσεις `strcmp` και `strncmp` επιστρέφουν την τιμή 1 αν τα δύο αλφαριθμητικά κείμενα που δίνονται στα ορίσματα των είναι ίσα, ενώ δίνουν την τιμή 0 αν διαφέρουν:

```
str1 = 'hello' ;
str2 = 'help' ;

strcmp (str1, str2)
ans =
    0
```

Η συνάρτηση `strncmp` συγκρίνει τους πρώτους *n* χαρακτήρες των δύο κειμένων. Για τις μεταβλητές του προηγούμενου παραδείγματος έχουμε:

```
strncmp(str1, str2, 2)
ans =
    1
```

Μπορούμε επίσης να συγκρίνουμε κάθε χαρακτήρα δύο αλφαριθμητικών κειμένων και να βρούμε ποιοι χαρακτήρες είναι ίδιοι και ποιοι διαφέρουν με την βοήθεια του τελεστή σχέσεων `==`. Ο τελεστής τοποθετεί στην αντίστοιχη θέση του πίνακα την τιμή 1 όταν οι χαρακτήρες είναι ίδιοι και την τιμή 0 στην περίπτωση που οι χαρακτήρες διαφέρουν.

```
A = 'fate' ;
B = 'cake' ;
A==B

ans =
    0 1 0 1
```

isletter, isspace

Οι συναρτήσεις `isletter`, `isspace` χαρακτηρίζουν τους χαρακτήρες ενός κειμένου και μπορούν να αναγνωρίσουν ποιοι από αυτούς είναι γράμματα και ποιοι από αυτούς είναι κενά, `tab` ή χαρακτήρας νέας γραμμής (`newline`):

```
mystring = 'Room 401' ;
```

```
isletter(mystring)
```

```
ans =
```

```
1 1 1 1 0 0 0 0
```

```
isspace(mystring)
```

```
ans =
```

```
0 0 0 0 1 0 0 0
```

14 ΠΟΛΥΔΙΑΣΤΑΤΟΙ ΠΙΝΑΚΕΣ

Για την χρησιμοποίηση δισδιάστατων πινάκων χρειαζόμαστε δύο δείκτες, ο πρώτος που αντιπροσωπεύει τις γραμμές του και ο δεύτερος που αντιπροσωπεύει τις στήλες του πίνακα. Οι πολυδιάστατοι πίνακες στην MATLAB είναι επέκταση των απλών δισδιάστατων πινάκων. Οι πολυδιάστατοι πίνακες χρησιμοποιούν περισσότερους από δύο δείκτες.

Για παράδειγμα ένας τρισδιάστατος πίνακας χρησιμοποιεί τρεις δείκτες, έτσι ώστε για να προσπελάσουμε το στοιχείο ενός τέτοιου πίνακα A που βρίσκεται στην τρίτη γραμμή, τέταρτη στήλη και δεύτερη σελίδα πρέπει να γράψουμε $A(3, 4, 2)$.

14.1 Δημιουργία Πολυδιάστατων Πινάκων

Μια μέθοδος δημιουργίας πολυδιάστατων πινάκων είναι η δημιουργία ενός απλού δισδιάστατου πίνακα ο οποίος στην συνέχεια αναπτύσσεται σε τρισδιάστατο με την βοήθεια των εντολών ομαδοποίησης πινάκων.

Για παράδειγμα έστω ο δισδιάστατος πίνακας A:

```
A = [5 7 8; 0 1 9; 4 3 6];
```

Για να προσθέσουμε και μια ακόμη διάσταση στον A χρησιμοποιούμε την εντολή

```
A(:, :, 2) = [1 0 4; 3 5 6; 9 8 7]
```

Το αποτέλεσμα θα είναι:

```
A(:, :, 1) =
```

```
5 7 8
0 1 9
4 3 6
```

```
A(:, :, 2) =
```

```
1 0 4
3 5 6
9 8 7
```

Με αυτό τον τρόπο μπορούμε να συνεχίσουμε να προσθέτουμε γραμμές, στήλες και σελίδες στον πίνακα τροποποιώντας τις διαστάσεις του.

Μια άλλη μέθοδος δημιουργίας πολυδιάστατων πινάκων είναι χρησιμοποιώντας το σύμβολο (:). Με αυτό τον τρόπο μπορούμε να τοποθετήσουμε σε μια ολόκληρη διάσταση ενός πίνακα μία αριθμητική τιμή:

```
A(:, :, 3) = 5
```

```
A(:, :, 3)
```

```
ans =
```

```
5 5 5
5 5 5
5 5 5
```

14.2 Δημιουργία πολυδιάστατων πινάκων με την βοήθεια συναρτήσεων

Οι συναρτήσεις `rand`, `ones`, `zeros` μπορούν να χρησιμοποιηθούν για την δημιουργία πολυδιάστατων πινάκων ακριβώς με τον ίδιο τρόπο με τον οποίο δημιουργούν δισδιάστατους πίνακες. Για παράδειγμα αν θέλουμε να δημιουργήσουμε έναν πίνακα 4x3x2 με τυχαίους αριθμούς, δίνουμε την εντολή:

```
B = randn(4,3,2)
```

repmat

Για να δημιουργήσουμε έναν πίνακα και να τοποθετήσουμε μια τιμή σε κάθε στοιχείο του πίνακα δίνουμε την εντολή `repmat`:

```
repmat (τιμή , [διασταση 1, διασταση 2, ..., διασταση n])
```

```
B = repmat(5,[3 4 2])
```

```
B(:, :, 1) =
5 5 5 5
5 5 5 5
5 5 5 5
```

```
B(:, :, 2) =
5 5 5 5
5 5 5 5
5 5 5 5
```

cat

Η συνάρτηση `cat` είναι ο πιο απλός τρόπος δημιουργίας πολυδιάστατων πινάκων από υπάρχοντες πίνακες. Ενώνει ένα πλήθος από πίνακες που έχουν ίδιες διαστάσεις:

```
B = cat(dim,A1,A2...)
```

Όπου `A1`, `A2....` είναι οι πίνακες που θέλουμε να ενώσουμε και `dim` είναι η διάσταση πάνω στην οποία θα γίνει η ενοποίηση.

14.3 Ανάκληση χαρακτηριστικών πολυδιάστατων πινάκων

Για να πάρουμε πληροφορίες που συσχετίζονται με τα χαρακτηριστικά πολυδιάστατων πινάκων η MATLAB διαθέτει συναρτήσεις οι σημαντικότερες των οποίων είναι οι ακόλουθες.

size

Η συνάρτηση `size` δίνει αναλυτικό πίνακα που περιέχει τις διαστάσεις ενός πολυδιάστατου πίνακα που δίνεται στο όρισμα της.

```
size(C)
ans =
    2 1 2 (γραμμές, στήλες, διάσταση3, διάσταση 4)
```

ndims

Η συνάρτηση `ndims` επιστρέφει τις διαστάσεις ενός πίνακα.

```
Ndims(C)
Ans =
    4
```

Για να χρησιμοποιήσουμε κάποιο συγκεκριμένο στοιχείο ενός πολυδιάστατου πίνακα μπορούμε να χρησιμοποιήσουμε τους ακέραιους δείκτες. Για παράδειγμα για να πάρουμε το στοιχείο ενός τρισδιάστατου πίνακα που βρίσκεται στην πρώτη γραμμή, Τρίτη στήλη, της δεύτερης σελίδας χρησιμοποιούμε το `C(1,2,2)`. Ομοίως και με τους δισδιάστατους πίνακες που αναφέρθηκαν σε προηγούμενο κεφάλαιο όλες οι εντολές που ισχύουν για την δεικτοδότηση μπορούν να εφαρμοστούν εύκολα και στους πολυδιάστατους πίνακες.

reshape

Αν δεν αλλάξουν οι διαστάσεις ενός πίνακα, η MATLAB διατηρεί τις διαστάσεις που έχουν οριστεί κατά την δημιουργία του πίνακα. Στην περίπτωση κατά την οποία ζητείται η αλλαγή των διαστάσεων του πίνακα μπορούμε να διατηρήσουμε τα περιεχόμενα του πίνακα με τη βοήθεια της εντολής `reshape`. Η σύνταξη της συνάρτησης είναι:

```
B = reshape (A, [s1 s2 s3 ... ])
```

Όπου τα `s1`, `s2`, ... αντιπροσωπεύουν τις διαστάσεις του νέου πίνακα B. Εδώ πρέπει να τονιστεί ότι ο νέος πίνακας θα πρέπει να έχει τον ίδιο αριθμό στοιχείων με τον αρχικό πίνακα. Η συνάρτηση `reshape` ανακατασκευάζει τον πίνακα ανά στήλες, δηλαδή φτιάχνει τον νέο πίνακα λαμβάνοντας διαδοχικά τα στοιχεία του παλιού πίνακα κατά στήλες.

14.4 Πολυδιάστατοι Πίνακες κειμένου

Όπως και στους πίνακες αριθμών, οι πολυδιάστατοι πίνακες κειμένου είναι επέκταση των δισδιάστατων πινάκων κειμένου. Για την κατασκευή τέτοιων πινάκων μπορεί να χρησιμοποιηθεί και η εντολή `cat` όπως και στην περίπτωση των πολυδιάστατων πινάκων αριθμών:

```
A{1,1} = [1 2 ; 4 5] ;
A{1,2} = 'Name' ;
A{2,1} = 2-4i ;
```

```
A{2,2} = 7 ;  
B{1,1} = 'Name2' ;  
B{1,2} = 3 ;  
B{2,1} = 0:1:3 ;  
B{2,2} = [4 5] ;  
C = cat (3,A,B) ;
```

15 ΑΡΧΕΙΑ

Η MATLAB έχει την δυνατότητα να μεταβάλλει το περιεχόμενο μεταβλητών της μεταφέροντας τα δεδομένα που περιέχονται σε αρχεία είτε αυτά περιέχουν δυαδικά είτε περιέχουν ASCII χαρακτήρες.

Ο κατάλογος των βασικών εντολών της MATLAB που χρησιμοποιούνται για την ανάγνωση και εγγραφή πληροφοριών από αρχεία είναι ο ακόλουθος

<i>Όνομα συνάρτησης</i>	<i>Συνοπτική περιγραφή εντολών</i>
fopen	Άνοιγμα αρχείου είτε για ανάγνωση είτε για εγγραφή δεδομένων
fclose	Κλείσιμο αρχείου
fread	Ανάγνωση δυαδικών δεδομένων από αρχείο και μεταφορά τους σε μεταβλητές του προγράμματος
fwrite	Εγγραφή σε αρχείο του περιεχομένου μεταβλητών και σταθερών με δυαδικό τρόπο
fscanf	Ανάγνωση ASCII δεδομένων από αρχείο και μεταφορά τους σε μεταβλητές του προγράμματος
fprintf	Εγγραφή σε αρχείο του περιεχομένου μεταβλητών και σταθερών σαν ASCII δεδομένα
gfetl	Ανάγνωση από αρχείο μιας γραμμής δεδομένων και αφαίρεση του χαρακτήρα της νέας γραμμής
fgets	Ανάγνωση από αρχείο μιας γραμμής δεδομένων με διατήρηση του χαρακτήρα της νέας γραμμής
sprintf	Εγγραφή σε μεταβλητή χαρακτήρων του περιεχομένου μεταβλητών και σταθερών σαν ASCII δεδομένα
sscanf	Ανάγνωση ASCII δεδομένων από μεταβλητή χαρακτήρων και μεταφορά τους σε μεταβλητές του προγράμματος
ferror	Συνάρτηση που επιστρέφει κωδικούς που συσχετίζονται με την κατάσταση εκτέλεσης εντολής εισόδου/εξόδου δεδομένων σε αρχείο
feof	Συνάρτηση που ελέγχει την ανάγνωση του κωδικού χαρακτήρα τέλους-του-αρχείου (end-of-file) στην προηγούμενη εντολή ανάγνωσης δεδομένων από το αρχείο
fseek	Μεταφέρει τον δείκτη ανάγνωσης/εγγραφής του αρχείου στην θέση που υποδεικνύεται από την εντολή
ftell	Συνάρτηση που επιστρέφει την θέση του δείκτη ανάγνωσης/εγγραφής του αρχείου
frewind	Μεταφέρει τον δείκτη ανάγνωσης/εγγραφής στην αρχή του αρχείου
tempdir	Επιστρέφει το πλήρες όνομα του καταλόγου στον οποίο το λειτουργικό σύστημα τοποθετεί τα προσωρινά αρχεία
tempname	Επιστρέφει το πλήρες όνομα ενός προσωρινού αρχείου το οποίο βρίσκεται στον κατάλογο των προσωρινών αρχείων

Στο επόμενο τμήμα αυτού του κεφαλαίου δίνεται αναλυτική περιγραφή της σύνταξης των εντολών όπως επίσης και αναλυτικά παραδείγματα της χρήσης αυτών στην μεταφορά δεδομένων από και προς τα αρχεία

15.1 Άνοιγμα και κλείσιμο αρχείων

Όταν θέλουμε να επεξεργαστούμε μεγάλο αριθμό δεδομένων τα αποθηκεύουμε σε αρχεία, και στην συνέχεια χρησιμοποιούμε ειδικές συναρτήσεις με τις οποίες μεταφέρουμε τα δεδομένα σε πίνακες της MATLAB για περαιτέρω επεξεργασία. Αν τα αποτελέσματα της επεξεργασίας είναι μεγάλος αριθμός δεδομένων τότε συνήθως τα εκτυπώνουμε ή τα αποθηκεύουμε σε αρχεία.

fopen

Απαραίτητη προϋπόθεση για την πραγματοποίηση οποιασδήποτε μεταφοράς δεδομένων από και προς αρχείο είναι η εκτέλεση της εντολής `fopen`. Με την εντολή αυτή πληροφορούμε τον διερμηνευτή (interpreter) της MATLAB για το όνομα του αρχείου που θέλουμε να ανοίξουμε, και τον τρόπο προσπέλασης των δεδομένων του.

Σύνταξη της εντολής:

```
<κωδικός αρχείου>
= fopen (<όνομα αρχείου>, <κωδικός προσπέλασης>) ;
```

όπου

<κωδικός αρχείου>, είναι μία ακέραια μεταβλητή ο αριθμός της οποίας πρέπει να αναφέρεται σε κάθε εντολή προσπέλασης του αρχείου.

<όνομα αρχείου>, είναι το όνομα του αρχείου το οποίο πρόκειται να ανοιχτεί, και

< κωδικός προσπέλασης >, είναι χαρακτήρες που δηλώνουν το είδος της προσπέλασης.

Επιπρόσθετοι κωδικοί προστίθενται σε αρχεία τα οποία βρίσκονται κάτω από τον έλεγχο του λειτουργικού συστήματος DOS, και MICROSOFT Windows. Οι κωδικοί αυτοί ορίζουν και τον τρόπο εγγραφής και ανάγνωσης των δεδομένων. Προσοχή θα πρέπει να δοθεί κατά την κατασκευή των προγραμμάτων. Δεν επιτρέπεται να ανοίξουμε το αρχείο σαν ASCII για διάβασμα δεδομένων ('rt') και στην συνέχεια να χρησιμοποιήσουμε την εντολή `fread()`.

Οι κωδικοί των τύπων αρχείου δεν είναι απαραίτητοι σε άλλα λειτουργικά συστήματα

Κωδικός προσπέλασης	Περιγραφή τρόπου προσπέλασης του αρχείου
r	Άνοιγμα αρχείου για ανάγνωση δεδομένων και τοποθέτηση του δείκτη ανάγνωσης στην αρχή του αρχείου
w	Άνοιγμα αρχείου για εγγραφή δεδομένων, διαγραφή των δεδομένων του, και τοποθέτηση του δείκτη εγγραφής στην αρχή του αρχείου
a	Άνοιγμα αρχείου για εγγραφή δεδομένων στο αρχείο και τοποθέτηση του δείκτη εγγραφής στο τέλος του αρχείου
r+	Άνοιγμα αρχείου για εγγραφή είτε ανάγνωση δεδομένων στο αρχείο και τοποθέτηση του δείκτη εγγραφής στην αρχή του αρχείου

Κωδικοί τύπων αρχείου	Περιγραφή τύπων αρχείου
t	Το αρχείο που πρόκειται να διαβαστεί περιέχει ASCII δεδομένα
b	Το αρχείο που πρόκειται να διαβαστεί περιέχει δυαδικά δεδομένα

Παραδείγματα:

```
fid = fopen( 'data.dat', 'rt' ) ;
.....
fid = fopen( 'd:\temp\data1.dat', 'r+b' ) ;
```

fclose

Μετά το τέλος της διαδικασίας μεταφοράς δεδομένων από και προς ένα αρχείο, είναι απαραίτητη η εκτέλεση της εντολής `fclose` η οποία μεταφέρει τα τελευταία δεδομένα και απελευθερώνει το αρχείο.

Σύνταξη της εντολής:

```
<κωδικός κατάστασης εντολής>
= fclose ( <κωδικός αρχείου> ) ;
```

όπου

<κωδικός αρχείου>, είναι η ακέραια μεταβλητή που αναφέρεται σε κάθε εντολή προσπέλασης του αρχείου.

<κωδικός κατάστασης εντολής>, είναι *a* ακέραιος αριθμός που επιστρέφει η συνάρτηση και έχει την τιμή 0 όταν το αρχείο κλειστεί σωστά ή λαμβάνει την τιμή -1 αν συμβεί οποιοδήποτε σφάλμα κατά την εκτέλεση της εντολής.

```
<κωδικός κατάστασης εντολής> = fclose ( 'all' ) ;
```

Η εντολή αυτή κλείνει όλα τα ανοιχτά αρχεία.

Παραδείγματα:

```
fid = fopen( 'data.dat', 'rt' ) ;
.....
st = fclose( fid ) ;
.....
st = fclose( 'all' ) ;
```

15.2 Εγγραφή και ανάγνωση δυαδικών δεδομένων από αρχεία

Η ανάγνωση και εγγραφή δυαδικών δεδομένων πραγματοποιούνται με την εντολή `fread()` και την `fwrite()`.

fread

Η εντολή `fread` μεταφέρει χωρίς να μετασχηματίσει τα δεδομένα του αρχείου σε μεταβλητή ή πίνακα της MATLAB. Το αρχείο μπορεί να έχει κατασκευαστεί και από πρόγραμμα άλλης γλώσσας προγραμματισμού αρκεί να ακολουθούν την ίδια δυαδική κωδικοποίηση αριθμών. Η εντολή `fread` χρησιμοποιείται μαζί με την εντολή `fwrite`.

Σύνταξη της εντολής:

```
[<μεταβλητή>, <αριθμός στοιχείων 1>]
= fread (<κωδικός αρχείου>, <αριθμός στοιχείων 2>,
<μήκος στοιχείου σε bytes>) ;
```

όπου

<μεταβλητή>, είναι η μεταβλητή στην οποία τοποθετείται η πληροφορία που διαβάζεται από τα αρχεία

<αριθμός στοιχείων 1 >, είναι ο αριθμός των στοιχείων που διαβάστηκαν

<κωδικός αρχείου>, είναι ο κωδικός αριθμός του αρχείου

<αριθμός στοιχείων 2>, είναι ο αριθμός των στοιχείων που πρέπει να διαβαστούν,

<μήκος στοιχείου σε bytes>, είναι ο αριθμός σε bytes που καταλαμβάνει η μονάδα των δεδομένων που διαβάζονται. Τα δεδομένα αυτά έχουν τυποποιηθεί και περιγράφονται στον πίνακα που ακολουθεί

Η τοποθέτηση της προδιαγραφής <αριθμός στοιχείων 2> δεν είναι υποχρεωτική. Όταν δεν τοποθετείται τότε διαβάζονται όλα τα δεδομένα μέχρι το τέλος του αρχείου.

Τυποποιημένο όνομα	Μέγεθος σε bits	Τύπος μεταβλητής
'char'	8	character
'uchar'	8	unsigned character
'schar'	8	signed character
'int8'	8	Integer
'int16'	16	Integer
'int32'	32	Integer
'int64'	64	Integer
'uint8'	8	Integer
'uint16'	16	Unsigned Integer
'uint32'	32	Unsigned Integer
'uint64'	64	Unsigned Integer
'float32'	32	Floating point
'float64'	64	Floating point
'float'	32	Floating point

'double'	64	Floating point
'short'	16	Integer
'long'	32	Integer

Παραδείγματα:

```
fid = fopen('data.dat', 'rb');
[A, c] = fread(fid, 100, 'int32');
```

Η εντολή `fread()` του παραδείγματος διαβάζει ακέραιους αριθμούς των 32 bits και τους τοποθετεί στον πίνακα A. Ο πραγματικός αριθμός των ακεραίων που διαβάζονται τελικά τοποθετείται στην μεταβλητή c. Συνεπώς η μεταβλητή c λαμβάνει τιμή που βρίσκεται ανάμεσα στο μηδέν και το 100.

```
fid = fopen('data1.dat', 'rb');
[Af, c] = fread(fid, 'float32');
```

Η εντολή `fread()` του παραδείγματος διαβάζει δεκαδικούς αριθμούς των 32 bits και τους τοποθετεί στον πίνακα Af. Ο πραγματικός αριθμός των αριθμών που διαβάζονται τοποθετείται στην μεταβλητή c.

fwrite

Η εντολή `fwrite` μεταφέρει χωρίς να μετασχηματίζει τα περιεχόμενα ενός πίνακα σε αρχείο.

Σύνταξη της εντολής:

```
<αριθμός στοιχείων>
= fwrite (<κωδικός αρχείου>, <μεταβλητή>,
<μήκος στοιχείου σε bytes>) ;
```

όπου

<αριθμός στοιχείων>, είναι ο αριθμός των στοιχείων που γράφτηκαν

<κωδικός αρχείου>, είναι ο κωδικός αριθμός του αρχείου

<μεταβλητή>, είναι η μεταβλητή από την οποία αντλείται η πληροφορία που γράφεται στο αρχείο

<μήκος στοιχείου σε bytes>, είναι ο αριθμός σε bytes που καταλαμβάνει η μονάδα των δεδομένων που διαβάζονται. Τα δεδομένα αυτά έχουν τυποποιηθεί και είναι τα ίδια που περιγράφονται στην εντολή `fread`.

Παραδείγματα:

```
fid = fopen('data.dat', 'wb');
a = rand(100,1) ;
c = fwrite(fid, 100, 'float');
fclose(fid) ;
```

Η εντολή `fwrite()` του παραδείγματος γράφει 100 δεκαδικούς τυχαίους αριθμούς των 32 bits από τον πίνακα A στο αρχείο 'data.dat'. Ο πραγματικός αριθμός των ακεραίων που τελικά γράφονται αποθηκεύεται στην μεταβλητή c.

15.3 Εγγραφή και ανάγνωση ASCII δεδομένων από αρχεία

Οι εντολές που μεταφέρουν ASCII δεδομένα από μεταβλητές της MATLAB σε αρχεία και το αντίθετο είναι οι `fgetl`, `fgets`, `fscanf`, `fprintf`.

Πρέπει να σημειωθεί ότι η μεταφορά δυαδικών αριθμητικών δεδομένων από και προς αρχεία είναι πολύ ταχύτερη από την μεταφορά των ASCII αριθμητικών δεδομένων και καταλαμβάνουν μικρότερο χώρο στον δίσκο. Γι'αυτό τον λόγο όταν θέλετε να αποθηκεύσετε μεγάλους αριθμητικούς πίνακες καλό είναι να δημιουργήσετε δυαδικά αρχεία.

fgetl

Η συνάρτηση `fgetl` μεταφέρει σε πίνακα χαρακτήρων το περιεχόμενο του αρχείου μέχρι να διαβαστεί ο χαρακτήρας `newline` ('\n'). Ο χαρακτήρας αυτός δεν τοποθετείται σαν τελευταίος χαρακτήρας στον πίνακα

Σύνταξη της εντολής:

```
<πίνακας χαρακτήρων> = fgetl (<κωδικός αρχείου>) ;
```

όπου

<πίνακας χαρακτήρων>, είναι ο πίνακας χαρακτήρων στον οποίο μεταφέρονται τα δεδομένα του αρχείου μέχρις ότου να συναντηθεί ο χαρακτήρας `newline`.

<κωδικός αρχείου>, είναι ο κωδικός αριθμός του αρχείου

Παραδείγματα:

```
fid = fopen('data.dat', 'rt');
String = fgetl(fid) ;
fclose(fid) ;
disp(String) ;
```

Η εντολή `fgetl()` του παραδείγματος διαβάζει την πρώτη γραμμή του αρχείου και την τυπώνει στην οθόνη του τερματικού.

fgets

Η συνάρτηση `fgets` μεταφέρει σε πίνακα χαρακτήρων το περιεχόμενο του αρχείου μέχρι να διαβαστεί ο χαρακτήρας `newline` ('\n'). Ο χαρακτήρας `newline` τοποθετείται σαν ο τελευταίος χαρακτήρας του πίνακα.

Σύνταξη της εντολής:

```
<πίνακας χαρακτήρων> = fgets (<κωδικός αρχείου>) ;
```

```
<πίνακας χαρακτήρων> =
fgets (<κωδικός αρχείου>, <αριθμός χαρακτήρων>) ;
```

όπου

<πίνακας χαρακτήρων>, είναι ο πίνακας χαρακτήρων στον οποίο μεταφέρονται τα δεδομένα του αρχείου μέχρις ότου να συναντηθεί ο χαρακτήρας `newline`.

<κωδικός αρχείου>, είναι ο κωδικός αριθμός του αρχείου,

<αριθμός χαρακτήρων>, είναι ο μέγιστος αριθμός χαρακτήρων που διαβάζονται αν δεν διαβαστεί πρώτα ο χαρακτήρας `newline`.

Παραδείγματα:

```
fid = fopen('data.dat', 'rt');
String = fgets(fid) ;
String = fgets(fid) ;
fclose(fid) ;
disp(String) ;
```

Η εντολή `fgets()` του παραδείγματος διαβάζει την πρώτη και την δεύτερη γραμμή του αρχείου και τυπώνει την δεύτερη γραμμή στην οθόνη του τερματικού.

fscanf

Η συνάρτηση `fscanf` μεταφέρει σε πίνακα το περιεχόμενο του αρχείου το οποίο είναι μορφοποιημένο με χαρακτήρες ASCII μέχρι να διαβαστούν όλα τα στοιχεία του προδιαγράψουμε.

Σύνταξη της εντολής:

```
[<μεταβλητή>, <αριθμός στοιχείων 1>] = fscanf
(<κωδικός αρχείου>, <κωδικός περιγραφής ASCII δεδομένων>,
<αριθμός στοιχείων 2> ) ;
```

```
[<μεταβλητή>, <αριθμός στοιχείων 1>] = fscanf
(<κωδικός αρχείου>, <κωδικός περιγραφής ASCII δεδομένων>,
[<αριθμός 1> <αριθμός 2>] ) ;
```

όπου

<μεταβλητή>, είναι η μεταβλητή στην οποία τοποθετείται η πληροφορία που διαβάζεται από τα αρχεία

<αριθμός στοιχείων 1 >, είναι ο αριθμός των στοιχείων που διαβάστηκαν

<κωδικός αρχείου>, είναι ο κωδικός αριθμός του αρχείου

<αριθμός στοιχείων 2>, είναι ο αριθμός των στοιχείων που πρέπει να διαβαστούν,

<κωδικός περιγραφής ASCII δεδομένων>, είναι κωδικός περιγραφής των ASCII δεδομένων του αρχείου. Η κωδικοποίηση είναι η ίδια που χρησιμοποιείται και στην γλώσσα C.

<αριθμός 1> και <αριθμός 2>, είναι δύο θετικοί ακέραιοι αριθμοί που περιγράφουν τις διαστάσεις ενός πίνακα, οι τιμές του οποίου θα διαβαστούν από το αρχείο

Παραδείγματα:

```
fid = fopen('data.dat', 'rt');
[a, c] = fscanf( fid, '%f' ) ;
fclose(fid) ;
```

Η εντολή `fscanf()` διαβάζει όλους τους δεκαδικούς αριθμούς του αρχείου `data.dat` και τους τοποθετεί στον πίνακα με το όνομα `a`. Ο αριθμός των δεδομένων που θα

διαβαστούν τοποθετείται στην μεταβλητή *c*, και εξαρτάται αποκλειστικά από το πλήθος των δεκαδικών αριθμών που έχουν γραφτεί στο αρχείο.

```
fid = fopen('data.dat', 'rt');
a = fscanf( fid, '%f', 100 ) ;
fclose(fid) ;
```

Η εντολή `fscanf()` διαβάζει τους πρώτους 100 δεκαδικούς αριθμούς του αρχείου `data.dat` και τους τοποθετεί στον πίνακα με το όνομα *a*.

```
fid = fopen('data.dat', 'rt');
a = fscanf( fid, '%d', [10 23] ) ;
fclose(fid) ;
```

Η εντολή `fscanf()` διαβάζει τους πρώτους 10*23 ακέραιους αριθμούς του αρχείου `data.dat` και τους τοποθετεί στον διδιάστατο πίνακα με το όνομα *a*.

fprintf

Η συνάρτηση `fprintf` μεταφέρει τα περιεχόμενα μιας μεταβλητής, σταθεράς ή πίνακα σε αρχείο. Τα δεδομένα μορφοποιούνται με χαρακτήρες ASCII..

Σύνταξη της εντολής:

```
[<αριθμός στοιχείων>] = fprintf
(<κωδικός αρχείου>, <κωδικός περιγραφής ASCII δεδομένων>,
<μεταβλητή> ) ;
```

όπου

<αριθμός στοιχείων>, είναι ο αριθμός των στοιχείων που γράφτηκαν στο αρχείο,

<κωδικός αρχείου>, είναι ο κωδικός αριθμός του αρχείου,

<κωδικός περιγραφής ASCII δεδομένων>, είναι κωδικός περιγραφής των ASCII δεδομένων του αρχείου. Η κωδικοποίηση είναι η ίδια που χρησιμοποιείται και στην γλώσσα C. Δόκιμοι κωδικοί χαρακτήρες είναι οι `d, i, o, u, x, X, f, e, E, g, G, c,` and `s`. Ειδικοί χαρακτήρες όπως οι `\n, \r, \t, \b, \f` χρησιμοποιούνται για να αλλάξουν γραμμή, να τοποθετήσουν τον χαρακτήρα carriage return, tab, backspace, and formfeed αντίστοιχα.

<μεταβλητή>, είναι η μεταβλητή το περιεχόμενο της οποίας μεταφέρεται στο αρχείο,

Παραδείγματα:

```
a = rand( 1, 100 ) ;
fid = fopen('data.dat', 'wt');
c = fprintf( fid, '%f\n', a ) ;
fclose(fid) ;
```

Η εντολή `fprintf()` μεταφέρει στο αρχείο `data.dat` όλους τους δεκαδικούς αριθμούς της μεταβλητής *a* μορφοποιημένους με χαρακτήρες ASCII. Αν όλα τα δεδομένα γραφτούν στο αρχείο τότε η μεταβλητή *c* θα λάβει την τιμή 100.

```
a = rand( 1, 100 ) ;
fid = fopen('data.dat', 'wt');
```

```
fprintf( fid, '%20.9f\n', a ) ;
fclose(fid) ;
```

Η εντολή `fprintf()` μεταφέρει στο αρχείο `data.dat` όλους τους δεκαδικούς αριθμούς της μεταβλητής `a` μορφοποιημένους με χαρακτήρες ASCII. Κάθε αριθμός του πίνακα καταλαμβάνει 20 χαρακτήρες από τους οποίους οι εννέα αφιερώνονται για να περιγράψουν τα δεκαδικά ψηφία. Κάθε αριθμός τοποθετείται σε μία γραμμή του αρχείου.

```
x = 0:.1:1;
y = [x; exp(x)];
fid = fopen('exp.dat','wt');
fprintf(fid,'%6.2f %12.8f\n',y(2:7,:));
fclose(fid);
```

Η εντολή `fprintf()` μεταφέρει στο αρχείο `exp.dat` τα περιεχόμενα του διδιάστατου πίνακα `y` από την γραμμή 2 έως την γραμμή 7. Τα δεδομένα κάθε γραμμής του πίνακα μορφοποιούνται βάσει των οδηγιών που δίνονται στην συμβολοσειρά: `'%6.2f %12.8f\n'`.

15.4 Διαχείριση σφαλμάτων εντολών προσπέλασης σε αρχεία

Τα σφάλματα που συμβαίνουν κατά την μεταφορά δεδομένων σε αρχεία έχουν να κάνουν με την συμπεριφορά του φυσικού μέσου αποθήκευσης (ο δίσκος δεν διαθέτει ελεύθερο χώρο ενώ εκτελούμε μία εντολή εγγραφής δεδομένων στον δίσκο, η μονάδα δίσκου έχει αποσυνδεθεί, χάλασε, η δισκέτα στην οποία γράφουμε ή διαβάζουμε δεδομένα δεν βρίσκεται στην μονάδα δισκέτας του υπολογιστή κοκ).

Η ανίχνευση σφαλμάτων προσπέλασης σε αρχεία πραγματοποιείται με την εντολή `ferror()`.

ferror

Η συνάρτηση `ferror` ανιχνεύει τα μηνύματα του λειτουργικού συστήματος σχετικά με την κατάσταση εκτέλεσης της τελευταίας εντολής προσπέλασης σε αρχείο.

Σύνταξη της εντολής:

```
<μήνυμα> = ferror(<κωδικός αρχείου>) ;
[<μήνυμα>, <αριθμός σφάλματος>] = ferror(<κωδικός αρχείου>) ;
```

όπου

<μήνυμα>, είναι μία ακολουθία χαρακτήρων στην Αγγλική γλώσσα που περιγράφει το είδος του σφάλματος που συνέβη,

<κωδικός αρχείου>, είναι ο κωδικός αριθμός του αρχείου,

<αριθμός σφάλματος>, είναι θετικός ακέραιος αριθμός ο οποίος αντιστοιχεί στο είδος του σφάλματος που συνέβη κατά την τελευταία εντολή προσπέλασης των δεδομένων του αρχείου.

Ο κωδικός αριθμός ορθής εκτέλεσης μιας εντολής προσπέλασης σε αρχείο είναι μηδέν.

Παραδείγματα:

```
fid = fopen('data.dat', 'wt');
[m,s] = ferror( fid ) ;
disp(m) ;
disp(s) ;
```

Η εντολή `ferror()` τοποθετεί στις μεταβλητές `m` και `s` το μήνυμα σφάλματος και τον κωδικό αριθμό του σφάλματος για την εντολή `fopen`. Στην συνέχεια τυπώνει το περιεχόμενο των μεταβλητών στην οθόνη.

15.5 Μετακίνηση του δείκτη προσπέλασης αρχείου

Οι εντολές που θα περιγραφούν σε αυτό το τμήμα των σημειώσεων περιγράφουν συναρτήσεις με τις οποίες μπορούμε να μετακινήσουμε τον δείκτη ανάγνωσης ή εγγραφής ενός αρχείου. Δείκτη ενός αρχείου εννοούμε την φυσική θέση στο αρχείο από το οποίο θα αρχίσει η εγγραφή ή ανάγνωση δεδομένων. Με την μετακίνηση του δείκτη του αρχείου έχουμε την δυνατότητα να προσπελάζουμε διαφορετικές περιοχές του αρχείου. Οι εντολές αυτές χρησιμοποιούνται συχνότερα με δυαδικά αρχεία.

feof

Η συνάρτηση `feof` ανιχνεύει αν στην τελευταία ανάγνωση δεδομένων διαβάστηκε και ο κωδικός χαρακτήρας `end-of-file` (το τέλος του αρχείου). Η συνάρτηση αυτή χρησιμοποιείται για να ελέγχουμε αν υπάρχουν διαθέσιμα δεδομένα για ανάγνωση. Το αρχείο πρέπει να έχει ανοιχτεί με μορφοποίηση ASCII χαρακτήρων.

Σύνταξη της εντολής:

```
<κωδικός αριθμός> = feof(<κωδικός αρχείου>) ;
```

όπου

<κωδικός αριθμός>, είναι 0 αν δεν έχει διαβαστεί ο χαρακτήρας `end-of-file` ενώ παίρνει την τιμή 1 αν κατά την ανάγνωση δεδομένων διαβαστεί και ο χαρακτήρας `end-of-file`,

<κωδικός αρχείου>, είναι ο κωδικός αριθμός του αρχείου,

Παραδείγματα:

```
fid = fopen('data.dat', 'rt');
while feof(fid) == 0
    s = fgetl( fid ) ;
    disp(s) ;
end ;
fclose(fid);
```

Η εντολή `feof()` ελέγχει αν έχει υπάρχουν διαθέσιμα δεδομένα στο αρχείο. Αν υπάρχουν εκτελούνται οι εντολές του `while` που διαβάζουν την επόμενη γραμμή δεδομένων και την τυπώνουν στην οθόνη. Όταν διαβαστεί ο χαρακτήρας `end-of-file` η εκτέλεση του βρόχου τερματίζεται.

fseek

Η συνάρτηση `fseek` μετακινεί τον δείκτη του αρχείου στην θέση που προδιαγράφεται. Η επόμενη εντολή προσπέλασης δεδομένων εκτελείται από την θέση που έχει τοποθετηθεί ο δείκτης.

Σύνταξη της εντολής:

```
<κωδικός αριθμός> = fseek(<κωδικός αρχείου>,
<σχετική θέση δείκτη>, <Θέση μέτρησης> ) ;
```

όπου

<κωδικός αριθμός>, είναι 0 αν η εντολή εκτελεστεί σωστά και λαμβάνει την τιμή -1 όταν η μετακίνηση στην επιθυμούμενη θέση δεν μπορεί να πραγματοποιηθεί,
 <κωδικός αρχείου>, είναι ο κωδικός αριθμός του αρχείου,
 <σχετική θέση δείκτη>, είναι η απόσταση σε bytes που πρέπει να τοποθετηθεί ο δείκτης του αρχείου θέτοντας σαν απόλυτη αρχή μετρήσεων την θέση που ορίζεται από την <Θέση μέτρησης>,
 <Θέση μέτρησης>, είναι ένας ακέραιος αριθμός που περιγράφει μία χαρακτηριστική θέση του αρχείου.

Οι χαρακτηριστικές θέσεις του αρχείου είναι τρεις:

```
'bof' ή -1, αρχή του αρχείου
'cof' ή 0, τρέχουσα θέση του αρχείου
'eof' ή 1, τέλος του αρχείου
```

Παραδείγματα:

```
fid = fopen('data.dat', 'rb');
s = fseek( fid, 80, -1 ) ;
if s == 0
    a=fread( fid, 100, 'float' ) ;
end
s = fseek( fid, -80, 'eof' ) ;
if s == 0
    ad=fread( fid, 10, 'double' ) ;
end
fclose(fid);
```

Η πρώτη εντολή `fseek()` μεταφέρει τον δείκτη του αρχείου στο 80ο byte από την αρχή του αρχείου και στην συνέχεια διαβάζει 100 δεκαδικούς αριθμούς. Στην συνέχεια μεταφέρει τον δείκτη του αρχείου 80 θέσεις πριν το τέλος του αρχείου και τοποθετεί στην μεταβλητή `ad` δέκα δεκαδικούς αριθμούς διπλής ακρίβειας. Προυπόθεση για την σωστή μεταφορά των αριθμών από το αρχείο στις μεταβλητές είναι το αρχείο `data.dat` να έχει κατασκευαστεί με τέτοιο τρόπο έτσι ώστε να περιέχονται τα σωστά δεδομένα στις θέσεις που διαβάζονται διότι πρέπει να γνωρίζουμε ότι η εντολή `fread` δεν κάνει έλεγχο αν τα δεδομένα που διαβάζει είναι σωστά.

ftell

Η συνάρτηση `ftell` επιστρέφει την τρέχουσα θέση του δείκτη μετρούμενη σε bytes από την αρχή του αρχείου.

Σύνταξη της εντολής:

`<θέση δείκτη> = ftell(<κωδικός αρχείου>) ;`

όπου

`<θέση δείκτη>`, είναι θετικός αριθμός που δηλώνει την θέση του δείκτη από την αρχή του αρχείου ή λαμβάνει την τιμή -1 αν η MATLAB δεν μπορεί να μετρήσει την θέση για οποιονδήποτε λόγο,

`<κωδικός αρχείου>`, είναι ο κωδικός αριθμός του αρχείου.

Παραδείγματα:

```
fid = fopen('data.dat', 'rt');
for i=1:10
    a=getl( fid ) ;
end
p = ftell( fid ) ;
disp(p) ;
```

Το πρόγραμμα του παραδείγματος διαβάζει τις πρώτες δέκα γραμμές του αρχείου και στην συνέχεια τυπώνει την τρέχουσα θέση του δείκτη του αρχείου, που βρίσκεται στην αρχή της ενδέκατης γραμμής του.

frewind

Η συνάρτηση `frewind` μεταφέρει τον δείκτη του αρχείου στην αρχή του. Η επόμενη εντολή θα διαβάσει ή θα γράψει δεδομένα από αυτή την θέση.

Σύνταξη της εντολής:

`frewind(<κωδικός αρχείου>) ;`

όπου

`<κωδικός αρχείου>`, είναι ο κωδικός αριθμός του αρχείου.

Παραδείγματα:

```
fid = fopen('data.dat', 'rt');
for i=1:100
    a=getl( fid ) ;
end
frewind(fid) ;
a1=getl( fid ) ;
a2=getl( fid ) ;
disp(a2) ;
disp(a1) ;
```

Το πρόγραμμα του παραδείγματος διαβάζει τις πρώτες εκατό γραμμές του αρχείου, στην συνέχεια ο δείκτης του αρχείου μεταφέρεται στην αρχή του, διαβάζονται οι πρώτες δύο γραμμές και στην συνέχεια τυπώνεται το περιεχόμενο της δεύτερης και κατόπιν της πρώτης γραμμής στην οθόνη του τερματικού.

16 ΕΥΡΕΤΗΡΙΟ

A

all, 58
ans, 15, 26, 27, 46, 47, 62, 63, 65, 66, 68, 69
any, 58

B

base2dec, 63
bin2dec, 63
blanks, 62
break, 33

C

cat, 13, 68, 69, 70
cell, 10, 15, 56
cellstr, 62, 65
char, 10, 15, 62, 63, 64, 65, 74
chol, 20
cholinc, 20
class, 63
clear, 61
computer, 15
cond, 20
condest, 20

D

dbclear, 43, 45
dbcont, 43, 44, 46
dbdown, 43, 45
dblank, 64
dblquad, 51
dbquit, 43, 45, 47
dbstack, 43, 45, 46
dbstatus, 43
dbstep, 43, 44, 46
dbstop, 43, 45
dbtype, 43, 45
dbup, 43, 45, 46
deblank, 62
dec2base, 63
dec2bin, 63
dec2hex, 63
det, 20, 28
double, 10, 11, 15, 62, 64, 75, 81

E

edit, 12, 54
eig, 20
eigs, 20
elfun, 5
elmat, 5
else, 30, 31, 57

elseif, 30, 31, 56
eps, 15
eval, 59, 62
expm, 21
ezplot, 51

F

fclose, 71, 73, 75, 76, 77, 78, 79, 80, 81
feof, 71, 80
ferror, 71, 79, 80
feval, 59
fgets, 48, 71, 76, 77
find, 58
findstr, 62
flops, 15
fmin, 51
fmins, 51
fopen, 49, 71, 72, 73, 75, 76, 77, 78, 79, 80, 81, 82
for, 12, 19, 20, 24, 30, 32, 33, 38, 42, 43, 45, 46, 47, 56, 57, 59, 60, 61, 82
format, 8, 11, 49
fplot, 51, 52, 53
fprintf, 71, 76, 78, 79
fread, 48, 71, 72, 74, 75, 81
frewind, 71, 82
fscanf, 48, 71, 76, 77, 78
fseek, 71, 81
ftell, 71, 82
fwrite, 49, 71, 74, 75
fzero, 51

G

gfttl, 48, 71

H

help, 34, 35, 55, 65
helpwin, 35
hess, 20
hex2dec, 63
hex2num, 63
humps, 51, 52, 53

I

if, 19, 30, 31, 43, 55, 57, 81
input, 60
int2str, 12, 59, 63
inv, 6, 20, 28, 29
iscellstr, 62
ischar, 62, 63
isinf, 58
isletter, 62, 65, 66
isnan, 58
isspace, 62, 65, 66

L

load, 10, 11, 12, 48, 49, 61
 lookfor, 34, 35, 55
 lower, 63

M

magic, 9, 15, 21, 34, 58
 mat2str, 63
 MATLAB, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
 18, 19, 20, 21, 23, 25, 27, 30, 31, 34, 35, 37, 38,
 44, 45, 48, 49, 51, 53, 54, 55, 56, 58, 59, 60, 61,
 62, 63, 65, 67, 68, 69, 71, 72, 74, 76, 82

N

nargin, 55, 56
 nargout, 55, 56, 57
 ndims, 69
 norm, 20, 26, 27, 34
 normest, 20, 34
 num2str, 63

O

ones, 68
 orth, 20

P

pascal, 21
 pause, 60
 pi, 15
 pinv, 20, 28, 29
 poly, 20
 polyeig, 20

Q

quad, 51
 quad8, 51
 qz, 20

R

rand, 21, 68, 75, 78, 79
 rank, 20, 29
 realmax, 15
 realmin, 15
 repmat, 68
 reshape, 69

S

save, 10, 11, 12, 49, 61
 schur, 20
 size, 69
 sparse, 15
 sprintf, 9, 63, 71
 sqrtm, 21
 sscanf, 63, 71
 startup.m, 14

str2num, 63
 strcat, 62
 strcmp, 62, 65
 strjust, 62
 strmatch, 62
 strncmp, 62, 65
 strep, 63
 strtok, 63
 struct, 10, 15
 strvcat, 62
 svd, 20
 switch, 19, 30, 31

T

tempdir, 71
 tempname, 71
 tic, 61
 toc, 61
 trace, 20

U

uint8, 15, 74
 upper, 63

V

varargin, 56
 varargout, 56
 version, 15

W

while, 19, 30, 32, 80

X

xor, 58

Z

zeros, 60, 68

A

Άδειοι πίνακες, 60
 Αντίστροφοι πίνακες, 28

B

ΒΟΗΘΕΙΑ, 34

A

Διόρθωση προγραμμάτων, 43

E

Ενοποίηση πινάκων, 16
 ενοποίησης πινάκων, 16

<i>M</i>	<i>Y</i>
Μέτρο διανυσμάτων, 26	Υποσυναρτήσεις, 57
Μοναδιαίος πίνακας, 25	
<i>Π</i>	<i>X</i>
παράθυρο εντολών, 6	χώρος εργασίας, 10
<i>T</i>	<i>Ψ</i>
τελεστές, 18, 19, 58	Ψευδοαντίστροφοι πίνακες, 28
τελεστές πράξεων, 18	

